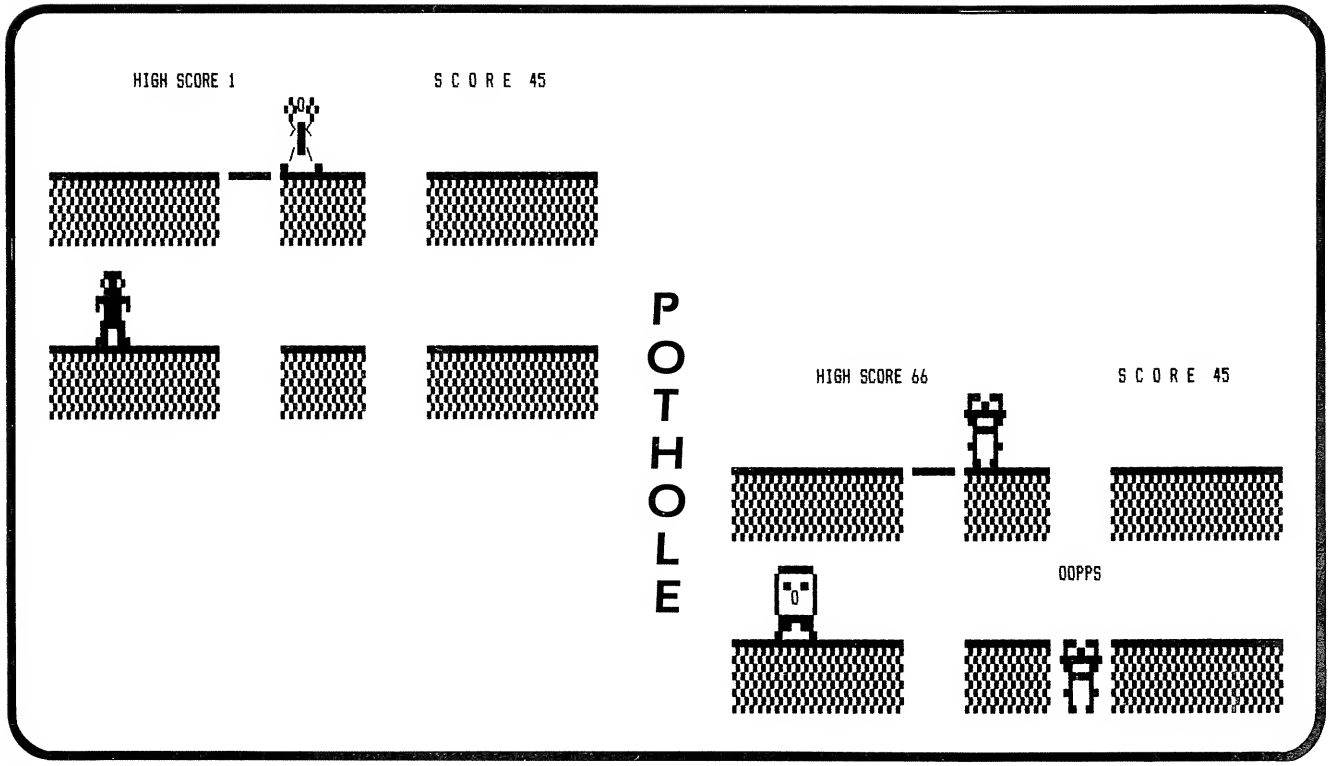
• TRS-80 • SYSTEM 80 • VIDEO GENIE
• PMC-80 • HITACHI PEACH
• TRS-80 COLOUR COMPUTER

Vol. 3, Issue 10, September 1982



Also in this issue:

PROGRAMMING:
Saving Disk Space

REVIEWS:
Visicalc
TC-8 Cassette Operating System

HARDWARE:
A Simple Joystick

SOFTWARE:
- PROPERTY INVESTMENT—Level II
- FASTER—Level II
- CRICKET—Level II
- CALENDAR—Colour
- HANGMAN—Colour
- SECTOR EDITOR—Peach

MICRO-80

***** ABOUT MICRO-80 *****

EDITOR:	RYSZARD WIWATOWSKI
ASSOCIATE EDITORS:	
SOFTWARE :	CHARLIE BARTLETT
HARDWARE :	EDWIN PAAY

MICRO-80 is an international magazine devoted to the Tandy TRS-80 Model I, Model III and Colour microcomputers, the Dick Smith System 80/Video Genie and the Hitachi Peach. It is available at the following prices:

	12 MONTH SUB.	SINGLE COPY
MAGAZINE ONLY	$ 26-00	$ 2-50
CASSETTE PLUS MAGAZINE	$ 65-00	$ 4-00 (cass. only)
DISK PLUS MAGAZINE	$125-00	$10-00 (disk only)

MICRO-80 is available in the United Kingdom from:
U.K. SUBSCRIPTION DEPT. 24 Woodhill Park, Pembury, Tunbridge Wells, KENT TN2 4NW

MAGAZINE ONLY	£ 16-00	£ 1-50
CASSETTE PLUS MAGAZINE	£ 43-60	£ N/A
DISK PLUS MAGAZINE	£ 75-00	£ N/A

MICRO-80 is available in New Zealand from:

MICRO PROCESSOR SERVICES, 940A Columbo Street, CHRISTCHURCH 1 N.Z. Ph 62894

MAGAZINE ONLY	NZ$ 43-00	NZ$ 4-00
CASSETTE PLUS MAGAZINE	NZ$ 89-00	NZ$ 5-00
DISK PLUS MAGAZINE	NZ$175-00	NZ$15-00

MICRO-80 is despatched from Australia by airmail to other countries at the following rates:

(12 MONTH SUB)	MAGAZINE	CASS + MAG	DISK + MAG
PAPUA NEW GUINEA	Aus$40-00	Aus$ 83-00	Aus$ 143-00
HONG KONG/SINGAPORE	Aus$44-00	Aus$ 88-00	Aus$ 148-00
INDIA/JAPAN	Aus$49-00	Aus$ 95-00	Aus$ 155-00
USA/MIDDLE EAST/CANADA	Aus$55-00	Aus$102-00	Aus$ 162-00

Special bulk purchase rates are also available to computer shops etc. Please use the form in this issue to order your copy or subscription.

The purpose of MICRO-80 is to publish software and other information to help you get the most from your TRS-80, System 80/Video Genie or Peach and its peripherals. MICRO-80 is in no way connected with any of the Tandy, Dick Smith or Hitachi organisations.

** WE WILL PAY YOU TO PUBLISH YOUR PROGRAMS **
Most of the information we publish is provided by our readers, to whom we pay royalties. An application form containing full details of how you can use your microcomputer to earn some extra income is included in every issue.

** CONTENT **
Each month we publish at least one applications program in BASIC for each of the microcomputers we support. We also publish Utility programs in BASIC and Machine Language. We publish articles on hardware modifications, constructional articles for useful peripherals, articles on programming techniques both in Assembly Language and BASIC, new product reviews for both hardware and software and we print letters to the Editor.

** LIABILITY **
The programs and other articles in MICRO-80 are published in good faith and we do our utmost to ensure that they function as described. However, no liability can be accepted for the failure of any program or other article to function satisfactorily or for any consequential damages arising from their use for any purpose whatsoever.

***** CONTENTS *****

MICRO-80 is registered by Australia Post - Publication SQB 2207 Category B

AUSTRALIAN OFFICE AND EDITOR: MICRO-80 P.O. Box 213, GOODWOOD SOUTH AUSTRALIA, 5034. TEL. (08) 211 7244

U.K. SUBSCRIPTION DEPT: 24 WOODHILL PARK, PEMBURY TUNBRIDGE WELLS, KENT TN2 4NW

Printed by: Shovel & Bull Printers, 379 South Road, MILE END 5031

Published in Australia by: MICRO-80 433 Morphett Street, Adelaide.

***** EDITORIAL *****

A new contender has thrown its hat into the ring - the Colour Genie is being imported into Australia by Bertas International and is already being advertised although delivery is not expected for some seven or eight weeks. The Colour Genie, unlike the Tandy Colour Computer, utilises the popular Z80 processor and features an extensive Microsoft BASIC in 16K of ROM. The basic Colour Genie features 16K of RAM (expandable to 32K), a true typewriter keyboard with user programmable function keys, a 40 x 24 line character display with both upper and lower case and programmable characters. The graphics screen features a resolution of 160 x 96 with up to eight colours and for $449 you get the base machine with a cassette recorder included. As well as these features, the Colour Genie includes a built-in RS232C port and can be expanded to a disk system through an expansion port and looks like good value for money. The Colour Genie will no doubt offer competition to the Tandy Colour Computer in the same way the System 80/Video Genie contend with the Model I/III.

The Tandy Colour Computer has even more competition in the U.K. where the Dragon 32, produced by Dragon Data, is selling so well that its makers expect to hold 5% of microcomputer market in the U.K. by the end of 1983. The Dragon is very similar to the Tandy, uses the Motorola 6809E processor, features the same text and graphic resolution, comes with 32K of memory but is priced below the Tandy. On the surface, the Dragon might appear to be a copy, but, although it uses the same hardware, it's an original design and, although both use an extended Microsoft BASIC, the routines are located in different places. Even so, about 80% of Tandy's software runs on the Dragon without modification.

While on the subject of colour computers, a problem has arisen in the preparation of programs for the Hitachi Peach. Although we have managed to convert the calendar program, unfortunately the Hangman program could not be completed in time for this issue so will be included in a later issue. Instead, we present the first submission by a Peach user - Disk Sector Editor. This program is modelled along the lines of Superzap for the TRS-80 which was also, originally, a BASIC program and should prove a useful utility to disk users.

Last time, I fear I did injustice to NEWDOS 80 V2.0 by stating that it provided no compatibility with Model III TRSDOS. This is, of course, incorrect as it does provide for the copying of files from Model III TRSDOS - if you read the documentation very carefully! The new Version 3.0 is not actually around the corner, but, in fact, half-way down the street and is not expected to be released for some six to nine months. However, as an interim measure, Apparat is offering optional hard disk support priced at an extra $US60 with Version 2.0 for the Model III hard disk drive, either the Tandy or the Apparat System.

- 0000000000 -

***** PEEKing (UK) - by Tony Edwards *****

Copyright and pirating is again in the news in the U.K. The Government have issued a Green Paper (a consultative document for discussion prior to statutory action) on the subject. This paper is a result of the Whitford Committee set up to review British copyright matters in 1973. The committee deliberated for four years before the Government finally issued the Green Paper on the subject. Such is the complexity of copyright matters in U.K. legislation. The Green Paper covers all aspects of copyright (including books, magazines and video/audio recording) but Chapter 8 relates specifically to programs stored in computers and on disks and tapes. It suggests that the protection should be "under the same conditions as literary works".

It will be a number of years before this is reflected in law but protection already exists in the form of the Forgery and Counterfeiting Act 1981 as this states that a person is guilty of forgery if he makes "use of a false instrument to defraud". A pirated copy of a program is a "false document" and anyone using it could be said to be defrauding the copyright owners of their rightful dues. This has not been tested in court yet with respect to computer programs, but it has been used against music pirates.

In the meantime, copying goes on at a very high rate amongst computer users and "clubs" have sprung up whose main aim is to copy programs from each other. This is possibly illegal, if the programs are copyright, but is it immoral? Typical games or minor utility programs cost £5 to £20 ($10 to $40) in the U.K. and the cost of the cassette they are carried on is only about 50 pence ($1). Hence the user sees a "profit" of some £20 per cassette sold. If a particular program sells thousands of copies a very large sum is available to be split between the programmers and the producers. With such a large difference between blank tape costs and taped program costs, there will always be a great temptation to make pirate copies. The major cause of pirate copying is the inflated cost of commercially available software and software producers should look to the problem faced by audio and video cassette producers, and reflect on the hard facts of life.

- 0000000000 -

***** INPUT/OUTPUT *****

From: Mr. E. Cruz - Matraville, N.S.W.

I have been experiencing a small problem with the System '80 I have (16K) and I wonder if you could offer me any advice. I have entered a program from a book put out by Creative Computing in the U.S.A. and one of the program lines calls for the use of 'DEF FNA(X)' and then a value. My system is indicating an error each time this program line is used and a phone call to Dick Smith Electronics has not helped as all they told me is that the System '80 will accept a 'DEF FN' instruction. A forward to the book by Creative Computing mentions that a few micro-computers will not accept this type of program instruction, and I wonder if you could advise whether, in your experience, the System '80 can accept a 'DEF FN' statement. Any advice on this problem would be greatly appreciated.

(Although mentioned in the BASIC manual, the DEF and FN reserved words are not supported in the Level II BASIC interpreter, even though they are parsed and tokenized by the interpreter. Their use in Level II programs will cause an SN error on the System '80 and an L3 error on the Tandy - a more descriptive error message.

This statement allows you to define a one line function which may be used elsewhere in the program and is available in Disk BASIC, which is provided with most Disk Operating Systems. Disk BASIC enhances the Level II BASIC by implementing those features of BASIC for which provision was made, but which were not included in Level II. Alternatively, Microsoft produce a separate program called 'Level III' which is intended for cassette systems and provides these enhancements and some others in place of Disk I/O features. - Ed.)

From: Mr. K. Bruce - Manly, N.S.W.

I would like to know how to lock out the RESET button on my TRS-80 Level II cassette based system. I know it can be done purely with software as I have the SARGON II chess program and pressing the RESET button simply vectors you to the start of the program.

I already know how to lock out 'BREAK' key, but I need to know the address of the vector for this Non Maskable Interrupt, so that I can change its contents to point me back to, if necessary, a machine language program to restore the stack, etc. and jump into into interpretation of my program where it left off.

Also being a believer of parallel reading, do you know where I could purchase a copy of "Pathways Through the ROM" - Softside Publications to supplement my IJG books. Thank you.

(The Reset button is, as you suggest, actually connected to the Non Maskable Interrupt (NMI) input to the processor and when pressed the processor resumes execution at location 66H (102 decimal). The code here is to check for the presence of a disk controller and if not found, as in a non-expanded cassette system, exits through the soft entry to BASIC via 66CH.

Before BASIC enters the immediate mode, if passes through a DOS exit that is normally used to force the 'RUN only' mode in Disk BASIC but in a Level II system it simply returns immediately. This is the exit that Sargon uses to return control to itself and is located at 41ACH (16812 decimal). You can regain control after the Reset button is pressed by loading a 'Jump' instruction to your program into locations 41ACH-41AEH.

As for "Pathways Through the ROM", I suggest you contact Softside Publications Inc. directly. Their address is:

6 South Street
Milford
New Hampshire 03055 U.S.A.

they will no doubt be glad to supply you with a copy. - Ed.)

From: Mr. D.H. Hebblethwaite - Lower Plenty, Vic.
JOYSTICK FOR THE SYSTEM-80

Some of your readers may be interested in the following modification I have made to my System 80 to add a joystick.

This modification to the System 80 allows the use of a switch type joystick (such as the Dick Smith Catalogue No. X-2020 without the need for a special interface. A six strand piece of ribbon cable is soldered to the DK3 (↑or ESC), DK4 (↓ or CTL), DK5 (←or B SP), DK6 (→or TAB), DK7 (Space Bar) and the AK6 (common) lines of the keyboard PCB . The other end is terminated in a 9-way DB socket mounted to the front of the bottom part of the System 80 case and into

which the joystick is plugged. The joystick switches therefore parallel the above-mentioned keys.

The modification costs little and works very well with PENETRATOR (which is compatible with the Alpha joystick in mode 5) but has not been tried on other games.

Finally as a System 80 owner I would like to compliment you on your efforts in publishing MICRO-80 which I find informative and useful.

(Thank you for your description of a simple Joystick for the System 80 and for your kind comments about the magazine. I have seen some such modifications and have tried it for myself and found it to work quite well with many games. Elsewhere in this issue is another approach that does not require the purchase of a commercial Joystick. - Ed.)

From: Robbie Thomas - Blakehurst, N.S.W

About six months ago I bought a copy of the program 'Asylum' and since then my family has been slowly going mad because we can not get past the second guard. If anyone can help me would they please tell me all the commands from when you are in the room to when you are past the guard.

From: Mr. B. Ronning - Lyneham, A.C.T.

I am a player of the TRS-80 game of Asylum. I'm writing to you in desperation because this infernal game is driving me insane. Please, do any of your staff know of a clue, book or anything available to help me?

Maybe you may know the answer to my problem yourself. I am at the part where I have just come out of the second maze and one inmate has a coat hanger that picks locks. I think I must have tried everything but still I cannot solve this part.

From: Mr. J. Terhoeve - Aitkenvale, Qld.

Could you please give me a small hint on the following:

How do you get out of the first maze in Asylum? Where to find the pick in the Dr. Who Adventure you have recently published.

If you have had no experience in these games, can I ask for some help from the readers anywhere out there?

(I'm afraid we can be of no help with any of these problems, but perhaps some other readers may be of assistance. - Ed.)

From: Dr. R. Langley - Balaclava, Vic.

The excellence of Ken Shillito's 1-line program LEMNISCATES, and of W. Eldridge's RESTORE LINE which can also fit into 1 line, both of which are reported in the March 1982 issue of MICRO-80, prompts me to disseminate another 1-liner - a utility written in Level 2 BASIC for 4K or more TRS-80 or System '80, which will MERGE 2 or more BASIC programs.

The listing below is an adamptation of a 7-line program by an unknown author in Micro-80, December 1979. Be careful to include the linefeeds (down arrows) exactly where shown. The program contains 255 characters, so from 240 onwards you must use X in Edit mode. Once a tape copy of the program is made, it can be CLOADed and RUN as usual, and it will then display brief operating instructions, including a pair of numbers to be POKEd from the keyboard, which cause the resident program to be released from below the floor of the BASIC program area. The first BASIC program to be merged is then CLOADed, followed by one or more others to be appended to each previous one's tail. A renumbering utility is often needed to make each appendage start with a larger line number than the previous program's last number.

Finally, the miniature utility line Ø is deleted before CSAVEing the merged programs.

```
Ø CLS:PRINT"CLOAD BASIC PROG TO BE MERGED
OPT RENUMBER
POKE16548,";PEEK(16548);":POKE16549,";PEEK(16549);"
RUN TO MERGE ANOTHER, ELSE DELETE Ø.
":A=17129:FORJ=1TO1:B=A:A=PEEK(B)+256*PEEK(B+1):J=-(A=0):NEXT:POKE16548,B-INT(B/256)*256:POKE16549,
INT(B/256):END
LINEFEEDS NEEDED AT END OF FIRST 4 "LINES".
```

(Thank you for your 1-line, Dr. Langley. I hope our readers can make use of it. - Ed.)

From: John Smith - Adelaide, S.A.,
VIDEO CONTRAST IMPROVEMENT - CHEAP AND SIMPLE!

One of the problems of the video screen is the reflection of stray light from the white phosphor powder. This reflection can be reduced greatly by a very simple method - placing ordinary black chiffon in front of the screen. The light generated by the tube is reduced a bit, but the brightness of the background is reduced much more, so the contrast is enhanced.

The only disadvantages are that the chiffon must occasionally be brushed clean, and that the visual angle is reduced more than with the green plastic filter. However, it is only at very wide angles of view that the problem arises. Two or three people sitting side by side have no problems.

First obtain a quarter of a metre of black chiffon from the "dress material" section of a department store. To make up the screen mount measure the dimensions of the "lip" in the housing around the tube. For the TRS-80 Model I the size is 266mm x 225mm, for Model II it is 295mm x 233mm, and Model III is 274mm x 212mm. For the Dick Smith video you will have to measure it for yourself! Now cut a piece of stiff card (or other suitable material) to the measured size, and round the corners to fit the video unit in front of the screen. Then cut out the centre leaving a frame 12mm wide.

Place this frame face down. Cut the chiffon somewhat larger than the frame, say 50mm to 80mm larger. Spread adhesive (plastic cement for instance) along a long edge of the frame, and place the edge of the chiffon on the adhesive starting at ONE CORNER. Stretch it VERY slightly and press it onto the adhesive leaving the overhang all at one end. When the adhesive has a grip, lift the chiffon from the short side where there is no overhang. Repeat the gluing procedure on this short side, and wait for the adhesive to grip.

You will now find that gentle pulling at the unglued corner of the chiffon makes it all tight. Fold the chiffon back, glue the remaining edges of the frame, gently pull the chiffon tight, lower it onto the glue and press it down. Wait for the adhesive to set, and trim the chiffon to the frame, with sharp scissors.

Turn on your computer, and when you have a display place your new filter screen in front. Voila! It's easier to see! (Note: the screen must be taped to the Model I as there is not enough lip to give support.)

(John made us a sample, and it really works. Readers who already have a green screen simulator might like to try stretching the chiffon over the outside of their simulator, it will cut down surface reflections enormously whilst still giving the pleasing green display. - Ed.)

FROM: Ray Holland, Mona Vale, N.S.W.

The listing below is a machine language driven program which I have developed to allow Tandy's Scripsit to drive a serial printer with support for handshake and automatic line feed. I have used this successfully with the Olympia ES-100P typewriter fitted with an RS-232 serial interface.

The routine was developed from information supplied by Tandy, Sydney. This routine is stored on the same disk as Scripsit and is the file which is called when an operator wants to use Scripsit with a serial printer. It automatically calls the Scripsit program, loads it and modifies it to use the driver and initialising routines which are part of this program.

In Line 450, the program tests for an input on the Tandy RS-232 interface pin 5 to test if the printer is busy. The handshake signal from the printer should be connected to pin 5 in order to indicate to the computer when to start or stop sending data.

At line 522, the program checks to see if the character is a "Carriage Return". If so, it will automatically add a Line-Feed command to each Carriage Return.

At Line 560 a printer initialisation routine can be installed. I have not required this for my application. However, you will notice spaces left for additional program steps if such

a routine is necessary.

(Thank you very much for this program, Ray. We have had enquiries from a number of readers wanting to use Scripsit with different printers. Ray's program should help those readers to make the necessary modifications. It should also be possible for System 80 users to make use of Scripsit in a similar manner. We would be happy to publish any adaptations of Ray's program which readers might develop for their own systems.

The listing below gives the machine code in the first two columns which may be entered directly into memory with a monitor, starting at BF00 Hex and the source code in the remaining columns. The source code may be entered into an Editor/Assembler. - Ed.)

```
                00100 ;THIS ROUTINE LOADS AND MODS SCRIPSIT TO PERMIT
                00110 ;USE OF OLYMPIA SERIAL PRINTER
                00120 ;
                00130 ;IT IS LOCATED IN HIGH RAM 'BF00' FOR 32K M/C
                00140 ;
BF00            00150           ORG     0BF00H
BF00            00160 SERIAL    EQU     $       ;ENTRY ADDRESS
BF00 E5         00170           PUSH    HL
BF01 215EBF     00180           LD      HL,SWITCH       ;TEST SWITCH FOR 1ST CALL
BF04 34         00190           INC     (HL)
BF05 35         00200           DEC     (HL)
BF06 E1         00210           POP     HL
BF07 2030       00220           JR      NZ,DRIVER       ;GO TO DRIVER IF NOT 1ST
BF09 115FBF     00230           LD      DE,DCBADR       ;DE= FILESPEC
BF0C CD3044     00240           CALL    LOAD            ;OPEN AND LOAD SCRIPSIT
BF0F CD2844     00250           CALL    CLOSE           ;CLOSE CMD FILE
BF12 3E01       00260           LD      A,1             ;SET 'SWITCH' TO SKIP
BF14 325EBF     00270           LD      (SWITCH),A      ;  THIS CODE FUTURE CALLS
BF17 3E21       00280           LD      A,21H           ;PATCH SCRIPSIT TO
BF19 326752     00290           LD      (5267H),A       ; PROTECT THIS DRIVEJR
BF1C 21FFBE     00300           LD      HL,SERIAL-1
BF1F 226852     00310           LD      (5268H),HL
BF22 3EC3       00320           LD      A,0C3H          ;PATCH SCRIPSIT TO EXECUTE
BF24 323966     00330           LD      (6639H),A       ; INIT ROUTINE
BF27 32565F     00340           LD      (5F56H),A       ; AND DRIVER ROUTINE
BF2A 2158BF     00350           LD      HL,INIT         ;ADDRESS OF INIT
BF2D 223A66     00360           LD      (663AH),HL      ; ROUTINE
BF30 2100BF     00370           LD      HL,SERIAL       ;ADDRESS OF DRIVER
BF33 22575F     00380           LD      (5F57H),HL      ; ROUTINE
BF36 C30052     00390           JP      5200H           ;ENTER SCRIPSIT
                00400 ;
BF39            00410 DRIVER    EQU     $
BF39 CD0460     00420           CALL    6004H           ;HALT IF CLEAR KEY HIT
BF3C 08         00430           EX      AF,AF'          ;GET CHARACTER INTO A
BF3D F5         00440 STORE     PUSH    AF              ;STORE CHARACTER
BF3E DBE8       00450 BUSY      IN      A,(0E8H)        ;INPUT CTS STATUS
BF40 CB7F       00460           BIT     7,A             ;TEST CTS
BF42 20FA       00470           JR      NZ,BUSY         ;LOOP IF CTS BUSY
BF44 DBEA       00480 STATIN    IN      A,(CNTREG)      ;TEST UART STATUS
BF46 CB77       00490           BIT     6,A             ;TEST FOR HIGH
BF48 28FA       00500           JR      Z,STATIN        ;IF NOT LOOP
BF4A F1         00510           POP     AF              ;RESTORES CHARACTER INTO A
BF4B D3EB       00520           OUT     (DTAREG),A      ;LOAD HOLD REG W/CHAR
BF4D FE0D       00522           CP      0DH             ;IS IT CARRIAGE RET
BF4F 2004       00524           JR      NZ,RETRN        ;RETURN IF NOT
BF51 3E0A       00526           LD      A,0AH           ;IF SO O/P LINE FEED
BF53 18E8       00528           JR      STORE
BF55 C3745F     00530 RETRN     JP      5F74H           ;GO BACK TO SCRIPSIT
                00540 ;
                00550 ;
BF58            00560 INIT      EQU     $
BF58 3E01       00570           LD      A,1             ;SET SCRIPSIT SWITCH FOR
BF5A 32627C     00580           LD      (7C62H),A       ; PRINTER INIT
                00590 ;
                00600 ;
                00610 ;
BF5D C9         00620           RET
                00630 ;
BF5E 00         00640 SWITCH    DEFB    0
BF5F 53         00650 DCBADR    DEFM    'SCRIPCIT/LC'   ;NOMINATE FILESPEC
BF6A 03         00660           DEFB    3               ;MARK END OF FILESPEC
00EA            00670 CNTREG    EQU     0EAH            ;IN READS UART STAT REG
                00680                                   ; OUT LOADS UART CTL REG
```

```
00EB        00690 DTAREG  EQU    0EBH          ;OUT LOADS UART THRE
            00700                              ; IN READS RX DATA
            00710 ;
            00720 ;
4430        00730 LOAD    EQU    4430H         ;CALL DOS TO LOAD FILE
4428        00740 CLOSE   EQU    4428H         ;CALL DOS TO CLOSE FILE
            00750 ;
BF00        00760         END    SERIAL
```

- 0000000000 -

***** MICROBUGS *****

Although we make every effort to ensure accuracy in the material we publish, inevitably errors and omissions will occur. In this section, we print corrections to those bugs that have been reported.

ADDENDUM TO LAST MONTH'S MICROBUG FOR MICRO GRAND PRIX - Vol.3., Issue 7, June 1982 (pp.25-29)

The alterations reported last month are only necessary for the System 80 as the two machines handle the 32 character screen mode differently and the program functions correctly on the Tandy, despite moving one excess character into screen memory.

***** FREE SOFTWARE LIBRARY - VOL. 1 *****

A number of people have contacted us regarding an error in the loading instructions for the program 'Level I in Level II'. According to the booklet, the name of the file on tape is noted as 'Level 1' on page 30. Actually, there are two different versions, one for the TRS-80 and the other for the System 80 with file names 'LV1TRS' and 'LV1SYS' respectively (as shown on page 4).

I have gotten into the habit of using only the first letter of the filename when loading SYSTEM tapes as I have found this to work and typing just 'L' would load the file with any of the filenames above. The SYSTEM tape loader does not require all the characters of the file name (up to six) to be keyed in but those that are, must correspond exactly with those read from tape - otherwise it will continue to search for the correct filename. So if you respond to the filename prompt with one character, only the first character of the filename read from the tape will be compared and the others skipped - but if you enter six characters as the name, then all six will be checked! I must admit that I didn't appreciate this myself until I looked at the problem more closely.

- 0000000000 -

***** READERS' REQUESTS *****

This column is a regular feature of MICRO-80. In it, we list all those articles, programs, etc. requested by our readers. We invite contributions from readers to satisfy these requests and will, of course, pay a publication fee for all articles, programs etc. printed. As a guide, we will pay a minimum publication fee of $10 for any article or review published. In the case of software reviews, we will aim to pay in accordance with the value of the program, up to a maximum of $25. So, if you write a good review which we publish and the usual selling price of the program in Australia is $19.95, then we would pay you $20. In that way, the successful reviewer will get the program he reviews, free. (Make sure you include the selling price in your review). Unfortunately, we cannot afford that policy on hardware (!) so we will pay in accordance with the merits of the review - generally of the order of $25. Submission of a review for publication automatically means that you are prepared to accept the figure we decide to pay you and no correspondence will be entered into. Payment will normally be made within 30 days of publication.

** ARTICLES **

- File handling on the '80
- Reviews of '80 compatible printers
- Reviews of commercially available software (including that produced by us!)
- Reviews of commercially available hardware
* A master index to the appropriate sections in the Tandy Manuals in Level I, Level II, DOS etc.
- Comparative reviews of disk drives
- An explanation of how to make full use of USR, PEEK and POKE statements
- Discussion of the various electric fields produced by the keyboard, tape recorder, monitor disk drives etc., how to measure them, how important they are and how to combat them

- The value of writing programs in straight BASIC

** SOFTWARE **

- Stock market program
* Horse racing system
* Morse code decoder
- Sub-routine Forum
- A new STAR-TREK game
- Programs of pharmaceutical interest such as Pharmacokinetics, Patient Medication Records, Drug information services etc.
- A "PRINT SCREEN" command for the Model 1
- A football game using graphics and sound
- Graphics car race as seen from the driver's position
- Tank game for a single player

** HARDWARE **
- Interfacing the '80 to external hardware
- Review on the performance of line filters
- Real Time clock
* Radio Teletype/Morse interfacing
- RFI (Radio Frequency Interference) suppression
- High speed card readers, interfacing and documentation for the Model II.

NOTE: An * denotes that we already have some suitable material on hand for this topic.

COLOUR COMPUTER OWNERS

If you would like some specific information in the form of an article, a software or hardware feature, write and let us know. We will include your request in this section in the hope another reader can contribute the information.

- 0000000000 -

***** SAVING DISK SPACE BY MERGING PROGRAMS OR ROUTINES - by B.H. Bussenschutt *****

Having filled most of my available disks and being short of the 'necessary' to buy more, I wondered how I could squash some of my programs up a bit. It then occurred to me that many of the small routines I'd saved for later use were each occupying the "minimum" disk space of 1.25K which is 1 granule and yet each routine was only a few hundred bytes long. Aha...the solution seemed near. I reckoned that if I could put several of these routines together in one program, and have a Menu to locate each one, I should be in business.

As always, Murphy's law applied ("What can go wrong - will"!!) All my routines started at line 10. Out with the manuals, and I discovered NAME (in DOS2.2 loose sheets) for renumbering programs, and MERGE for adding programs together. The only tricky bits are:

(a) MERGE only works when the program on disk (which is going to be added to the one in memory) has been saved with the ,A option - i.e. saved in ASCII format (Read the fine print under MERGE on page 7-29 and 7-30 of the Tandy DOS manual).

(b) NAME only works under BASICR. This variation of BASIC is also explained in the DOS2.2 loose sheets.

The resulting process came out as follows:

1. Set up a MENU at the start so you can find them when they are merged.

```
e.g. 10 CLS:'PROGNAME "VARIOUS/TXT".
     20 PRINT @ 200, "M E N U":PRINT.
     30 PRINTTAB (14) "1. program one".
     40 PRINTTAB (14) "2. program two".
     50 PRINTTAB (14) "3. program three".
     60 PRINTTAB (14) "4. program four".
     70 PRINTTAB (14) "5. program five".
     80 PRINTTAB (20) "";:INPUT "SELECT PROGRAM NO. ";P
     90 ON P GOTO 200, 400, 600, 800, 1000
    100 END
```

2. Save it with the ,A option, e.g. SAVE "VARIOUS/TXT:1",A

3. If you haven't already done it, load BASICR. (Note BASICR, not BASIC)

4. Answer Files & Memory Size questions normally.

5. LOAD "program one".

6. Renumber the program to start at 200, using...NAME 200, old startline, proposed increment. If program one has less than 20 lines, proposed increment can be 10, otherwise use a lesser increment number, or a greater number spacing in line 90 above.

7. Merge the resident program with the ASCII saved part, e.g. MERGE "VARIOUS/TXT:1".

8. If you want to check that it happened, LIST it and see.

9. Save it again using the ,A option, as 2 above.

10. Repeat lines 5 to 9, with Programs two, three, four, etc., but using 400, 600, 800 etc. instead of 200 in line 6.

11. When complete, check that each segment has an END, or else a way to return to the MENU. If not, EDIT it, to include these.

12. When complete, change the name to /BAS using the DOS function RENAME - e.g. RENAME VARIOUS/TXT:1 TO VARIOUS/BAS.

- 0000000000 -

***** REVIEW OF THE DATESTONES OF RYN - a Dunjonquest program. By Mark Bell *****

Dunjonquest is a series of fantasy role playing games by Automated Simulations. This series contains two types of program, the Dunjonquest Epic, and the Dunjonquest Microquest. I own a Dunjonquest Microquest. The series runs on the same theme in both program types, but I have not played a Dunjonquest Epic and thus cannot comment on it.

In Dunjonquest (as in other role playing games) you are assigned a previously designed character, who is your alter ego for the game. This character is made up of a number of characteristics. In Dunjonquest there are six, and each is rated numerically. The higher the number, the greater/stronger the characteristic. These ratings do not change during the course of the game, though they may differ from character to character. In a Microquest you have a single character who never changes, while in a Dunjonquest Epic, this character is variable.

In the usual role playing game you direct your character through an imaginary world which has been designed by someone else, who is the referee for the game. In a Dunjonquest game, the computer is the referee and the world you enter is in its memory. In Dunjonquest, the fantasy world entered is a complex of rooms and caverns, commonly called a dungeon. (That's the way the Oxford Dictionary spells it, so that's the way I'm going to spell it).

The Microquest I own is the second in the Dunjonquest series, and is the first Microquest. It is called "The Datestones of Ryn". In this program, your (character's) mission is to enter a cavern complex and rescue a number of valuable datestones from the hands of a band of cut-throates, thieves and robbers. You must enter the complex alone, and armed with your sword, shield and bow, try to retrieve as many datestones as possible before the time limit expires.

Within the complex there are various methods of coming to a sticky end. These usually involve being dispatched at the sword or dagger of a robber, but there are other nasties that live in the shadows which you must vanquish in order to achieve your goal.

The section of the dungeon you are currently in, and any monsters/robbers and datestones present are shown graphically on the screen, along with several status readouts. The game is played in "real time" with thirteen single key commands (for example, 'A' for Attack and 'L' for Turn Left), plus numbers nought to nine for movement.

The game comes on cassette with an eighteen page instruction booklet for TRS-80 16K Level II. There are also instructions for putting the game on disk. On side one of the cassette is the BASIC (with machine language subroutines) program, which must be loaded after a memory size has been set. On side two is a data file which is loaded by the BASIC program every time the game is played.

I enjoy this game very much. But, at first, it was extremely hard to load, and there are some bugs in it. The most irritating of these is that if you attempt to enter a particular passage in the complex, the program crashes. What is beyond this passage is unknown; there may even be a whole section of the complex I have never seen! These problems may only occur in my copy of the program, but it has made me uneasy about buying further programs in the Dunjonquest series.

I would be grateful if other owners of Dunjonquest programs would contact me with details of any problems they have or have not had.

- 0000000000 -

***** VISICALC - A Software Review by A.G. Hudson *****

VISICALC was available for APPLE and PET and one was hearing good things about it before the version for the TRS-80 Model 1 arrived.

A Machine Code Disk Program loading and running directly under TRSDOS, this is the sort of program you can forget about as soon as you have loaded it. From then on, you use the screen to look at a huge sheet of paper divided into 255 rows and 55 columns, or such lesser numbers as you require. Of course, you cannot see all of this at once, but can place a cursor (flashing or still) on the row and column you want, either by entering co-ordinates for the column and row (e.g. A 3) or by moving the cursor with the arrow keys. Holding down the arrow keys gives automatic repetition for longer movements.

The column and row references are displayed along the top and left hand edges of a window normally covering six columns and twelve rows. The window can be divided vertically or horizontally so that two distant parts of the sheet can be viewed simultaneously. The sheet scrolls automatically behind the window(s) horizontally or vertically as the cursor is moved, and when the window is split, scrolling can be synchronised or not as required.

The procedure is to position the cursor at the point where entry is to be made. There are then three choices - one can enter a label (usually in the first row to label the columns and in the first column to label the rows); one can enter a value or one can enter a calculation which can be based on the figures already entered simply by referring to the place where the entry has been made by its column and row co-ordinator. These may be read into the formula by picking them out with the cursor.

The normal arithmetic functions are supported, including exponentiation. In addition, one has ABS, INT, SQRT, LN, EXP and LOG10, with the trigonometric functions (done in radians) SIN, COS, TAN, ASIN, ACOS and ATAN. All these are preceded by the @ key which calls all functions except the basic arithmetical ones.

It is also easy to COUNT the number of entries in a list or to SUM or AVERAGE them. MAX and MIN will calculate the respective values in the list while LOOKUP will find the corresponding value in the next column or row to the nearest entry below or equal to the value specified.

More complicated to me, though easy enough to VISICALC, is the calculation of the net present value of cash flows entered into a range.

Three lines above the window are reserved to tell you what you are doing and also the entry you are in process of making.

Calculations set up on one row or column can be repeated on others without re-entering all the expression again.

Simple control instructions permit variation of the width of columns, fixing of title areas, deletion or insertion of rows of columns, various formatting of entries, replicating commands and printing.

When all is ready, changing a single entry can be followed by automatic recalculation of every figure affected by the change.

Further commands allow formatting of individual locations, rows or columns (by replication) or of the whole area. Entries can be left or right justified or set as decimal or money, or even as a row of asterisks to form a symbol bar chart.

The print command adopts the current position of the cursor as the top left hand corner of the area to be printed, and on entry of the bottom right corner, the printer prints everything in the block thus marked out. Provided the number of column is within the line length available on the printer, the format will be maintained also. For this, the ability to print 132 characters to the line can be useful.

The whole can be saved to disk complete and loaded from disk or, alternatively, some part (defined by upper left and lower right) positions can be saved to disk. The files thus saved can be loaded, but when this is done, the file loaded overlays anything currently in memory. Thus, for instance, a single column of figures can be saved from one sheet and used as an entry onto another.

With all these possibilities, the documentation needs to be good, and it is. A series of exercises takes the learner through every command to demonstrate how it is used and the handbook concludes with a first-class summary of them all.

Snags? I have found just one minor one. The print command allows for typing in some sort of title or heading, but whatever is typed in at this stage goes straight into the printer's buffer and does not appear on the screen. Moreover, it cannot be corrected, even if you realise you have made a mistake. Anywhere else, correction is simply a matter of positioning the cursor over the erroneous entry and typing in the correct one.

One way in which I have used VISICALC is to create a set of blank forms with all the computation that will be required built into them. Then the blank form is loaded from disk, the data is entered into the appropriate places and 'hey presto' the totals, sub-totals, percentages, ratios and differences all appear in their allotted places ready for printing.

VISICALC is likely to be a very good buy for all serious users of the TRS-80.

- 0000000000 -

***** DUPLITAPE - A Software Review by Dave Futcher *****

DUPLITAPE, written by John Newgas and published by ALBION SOFTWARE, London, is an excellent real time tape copying program for use on the TRS-80 or the VIDEO GENIE. DUPLITAPE offers the facilities in one program to copy / backup any 500 baud tape of many different types including BASIC programs, machine code programs, data tapes, Electric Pencil files.

In addition, DUPLITAPE works with protected tapes and programs that run immediately on loading. What is particularly valuable is that you can backup a whole mixture of programs of different types mixed on one cassette. Also very important is that the computer memory size is not important, as the amount of available RAM does not limit the size of file which is being copied. This means you can use a 4K machine to copy a full 16K program. This is where DUPLITAPE is a winner - the program works where before there was not enough room for the copying program and the program being copied to both fit into memory.

How then do you do all this? And what do you need? All you need is a computer, two tape recorders, a blank cassette, and DUPLITAPE.

The two cassettes are connected to the computer by means of the normal computer cassette lead and the black jack going to one cassette and the grey jack going to the other cassette. One cassette is set in the record mode with a blank cassette, and the other in the play mode with the tape to be copied.

While operating, DUPLITAPE gives on screen instructions and displays a monitor window which shows moving graphics when DUPLITAPE is actually copying.

There have been many copying/backup programs for the TRS-80 and I expect that most owners have got a few already. But DUPLITAPE has something else to offer - we all know the programs that we have been unable to copy - so far, DUPLITAPE has not failed me. DUPLITAPE is a welcome acquisition to my library of programs.

The program is supplied on cassette and it comes with an excellent instruction booklet which not only tells you how to run and use the program, but also it explains how and why DUPLITAPE works. I found the instructions clear and easy to understand, although a diagram showing how to connect the machines would have been useful.

DUPLITAPE is well worth the £9.50 charged. All those 'protected' programs can now be backed up, as can those long Adventures. But I particularly like being able to backup my general data tapes, and to copy long and mixed tapes. Well done, ALBION, it's a great program!!

- 0000000000 -

***** BATTLEGROUND - Software Review by M.H. Dachsler *****

Battleground , by Instant Software Inc., is available on tape for 16K Level II machines and is written in BASIC.

A two-player game, this tape programs the computer to act as a simple map board, a communications centre for your troops, and an umpire to evaluate the battle after playing your number of moves (between 2 and 19).

Battleground allows each player (German and American) four turns per round to fire artillery, drive and shoot tanks, send infantry into attack, task engineers to blow obstacles, and fly aircraft to bomb or parachute troops into enemy territory after each player has chosen the content of his force and located it. In addition to the above, you initially choose vehicles, mines, blocks and bunkers up to specified values. A player then locates his forces, the screen clears and his opponent locates his. The screen then clears again and redraws the battlefield with both forces positioned and no mines showing. As the game progresses, the computer resolves all actions by deducting points from a combat value for each item until that is less than zero, and they are removed from the 10 x 20 (200 squares) grid battlefield.

After the pre-determined number of moves (give or take up to two), or calling for the game to end, "The Generals" evaluate the actions and give a very fair result in terms of advances by each side intact forces left and the magnitude (including a numerical value) of one force's win over the other, or a draw situation.

This is a very good value program which loads easily, provides good input prompts which return if any entry is faulty. Five and a half pages of instructions may look overwhelming at first, but playing is simple and there is room to get quite involved in a best strategy.

As a relatively inexpensive war game which should suit all ages, I found Battleground to be just what my war-mongering son wanted, and I grew to like it very quickly. Perhaps he will beat me next game!

- 0000000000 -

***** THE J.P.C. PRODUCTS TC-8 CASSETTE OPERATION SYSTEM - A Review by I.H. Robertson *****

Before purchasing my TRS-80 I read quite a few articles in an attempt to establish the most suitable set-up for my requirements. As it turned out, I ended up with a Model 1, expansion interface, Sanyo green screen monitor and an Olivetti ET 121 daisywheel typewriter/printer. Lack of funds prevented the purchase of a pair of disk drives which would have allowed me to complete the system nicely.

Thus, I was stuck with the frustrations of a tape based system which has cost me dearly in time and effort. Being basically lazy, I hate having to put in more work and time than necessary and so began the search for a more reliable, faster method of data storage and retrieval which would bridge the gap until I could afford better things.

My first attempt to conquer the problem was to race out and purchase a copy of "HI-SPED" and, although this program did speed up operation to some extent, I was still not happy as the new ROMs in my machine would not allow an increase beyond approximately 800 baud. Nor could I afford a stringy floppy, especially as it is my aim to eventually go the whole hog with disks.

An American magazine supplied the solution in the form of an advertisement for the TC-8. I immediately contacted J.P.C. Products in the States and, after two weeks, the reply came back saying that the interface was available in Australia from De-Forest Software of Nunawading, Vic. A quick telephone call confirmed that the TC-8 was indeed available and "would I like the kit for $120 or the fully built version for $150?" Despite my lack of previous kit building experience, and to satisfy "Scrooge" who was sitting heavily on my shoulder at the time, I ordered the kit.

Seven days later it arrived through the post, well packed and in good condition. Immediately after work I raced home, unpacked it, and sat down and read the assembly instruction manual from start to finish. And what a good manual it is, with very well laid out instructions and easy to follow steps. Assembly was particularly simple, but to make sure everything went well, I took every care possible and did not hurry - it took just on four hours which I consider acceptable for a person of my experience (nil!).

Following completion, it is necessary to connect the ribbon cable to the CPU or E/I and plug in the supplied power pack. At first this appeared easy, but unfortunately the power pack was supplied without a jack plug for connection to the TC-8, or any instructions for fitting one - just great if you are a novice like myself who can't tell the positive wire from the negative one when both are unmarked except for a white stripe on one! Luckily I have a friendly electrical store owner who fixed the problem, but it could be a big let down for anybody who is stuck out in the bush 100km. from the nearest town.

Following power-up, I was immediately rewarded with a sense of satisfaction and achievement. The TC-8 worked first time! - My labours had paid off. Now I had to familiarise myself with the operation of the system and even this is very easy once you get to know the few additional commands now available through the supplied software program "UTIL". These are divided into two separate sets, those that work on ordinary "BASIC" tapes and those that are used on "SYSTEM" tapes. The new commands are:-

SAVE - Equivalent to "CSAVE". However, you can now use up to 8 characters to identify program name.

LOAD - Equivalent to "CLOAD".

LOAD? - Equivalent to "CLOAD?".

LOADN - Displays list of programs on screen. This command can be very useful for exactly positioning a tape at the end of any program if you wish to write more programs on the tape.

KILL - Kills "UTIL" program so that you can use that portion of memory previously occupied if your BASIC program is large enough to require all your memory.

RSET - If you are operating an unmodified CTR-41 this allows you to use the machine's controls without removing the remote plug. A CTR-80 does not require this command.

RUN - Used in conjunction with a file name, this command automatically loads and runs your program.

PUT - SYSTEM type command for saving an M/L program - e.g. PUT"SCRIPSIT"4300,69C5,4300.

GET - SYSTEM equivalent to LOAD.

GET? - SYSTEM equivalent to LOAD?. Compares and verifies tape with memory.

GETN - SYSTEM equivalent to LOADN.

OPEN - A BASIC statement required to OPEN file before execution of a PRINT# or INPUT# statement.

CLOSE - A BASIC statement required to CLOSE file after execution of a PRINT# or INPUT# statement.

PRINT AND INPUT# - The equivalent of normal TRS-80 BASIC statement but the minus ("-") sign is omitted. These statements support multiple variables in a line but, with very efficient packing, there is minimal advantage to be gained. The only restrictions seem to be that string length is confined to 127 characters and commas are not allowed.

In addition, a small monitor appropriately named "TINY" is included in the software package, and this allows automatic relocation of "UTIL", identification of "SYSTEM" and TC-8 programs, examination of memory locations, writing backup copies of "UTIL", and jump to a specified HEX location.

Well, after all that, how does it work? If you are looking for a guaranteed way of saving time without laying out big dollars, this has to be the way to go! For example, one program I have been developing was taking approximately 3.5 mins. to CSAVE and CLOAD? each way which, over 1 or 2 nights, can run into quite a lot of time if you are in the habit of regularly CSAVEing. It now takes around 40 seconds.

Each way - and I reckon that 80 - 90 seconds total time is a heck of a lot better than 7 or 8 minutes (or 2 cups of coffee!!)

There are also huge improvements to be had if you are using your '80 for tape data I/O. As an extra bonus, the TC-8 does not rely on the cassette relay to control your cassette recorder which greatly reduces the risk of relay microwelding.

As can be seen from the above commands, the TC-8 can also be used for SYSTEM tapes and again, gives excellent results. However, in the short time I have been using it, I have noticed that the control program "UTIL" will not always remain transparent, with the result that there can be some clashes - e.g. "LEVEL III" will load but not run with "UTIL". A similar situation exists with "SCRIPSIT", but "PACKER" appears to be unaffected when called upon to do its duty. By trial and error I have worked out that those programs which look at the keyboard seem to require the killing of "UTIL" which is a habit I have adopted whenever they are being used.

At this time, the only real problem I have found with the existing system is the inconvenience of having to keep swapping the cassette connecting cable from the keyboard to the TC-8 interface unit. However, once again my friendly electrical store owner has come to the rescue with an idea for a switching unit which will allow permanent connection to both keyboard and TC-8. I hope to have it installed shortly but, in the meantime, I am quite willing to put up with this small inconvenience in exchange for the time-saving and confidence the TC-8 affords.

In conclusion, I have found the TC-8 and accompanying software to be extremely useful and reliable. Although I have rarely experienced loading or saving problems with my TRS-80, I have read enough articles on the cassette system to have doubts about the long term reliability of the standard set-up and I have spent many frustrating hours waiting for completion of cassette loading, saving or verifying. For $120, the TC-8 kit offers ease of assembly (mine worked first time!), reliability and confidence (I haven't yet lost a byte and volume setting seems fine if set anywhere between 1 and 8!), speed (claimed to be 5 times faster than the normal 500 baud!), inbuilt keyboard debounce and very, very good documentation. A system worth considering if you feel the way I once did.

- 0000000000 -

***** ADD A JOYSTICK TO YOUR '80 - by R. Brown *****

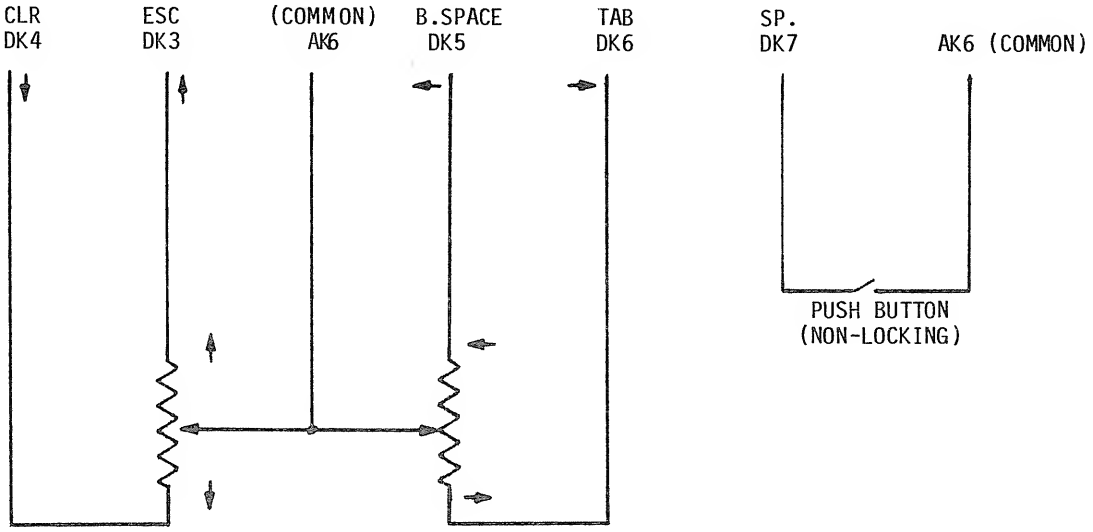
A simple and inexpensive method of interfacing a joystick to the '80 computer.

The '80 uses software scanning of all keys. By intercepting some of these keys and transferring them to a joystick, we achieve for about $10 what would normally cost a great deal more.

Most games use the arrow keys and space bar on the '80 computers. These keys have the same address line (AK6), but are wired to different data lines (D3-D7). Using 6 wires (1 for each data lead, and 1 for AK6), and terminating these to a 7 pin DIN plug, we intercept the keyboard.

Looking closely at the underside of the keyboard unit, follow the tracks around the various keys, locating the common which runs between the arrow and space keys. This common will become the common on the joystick.

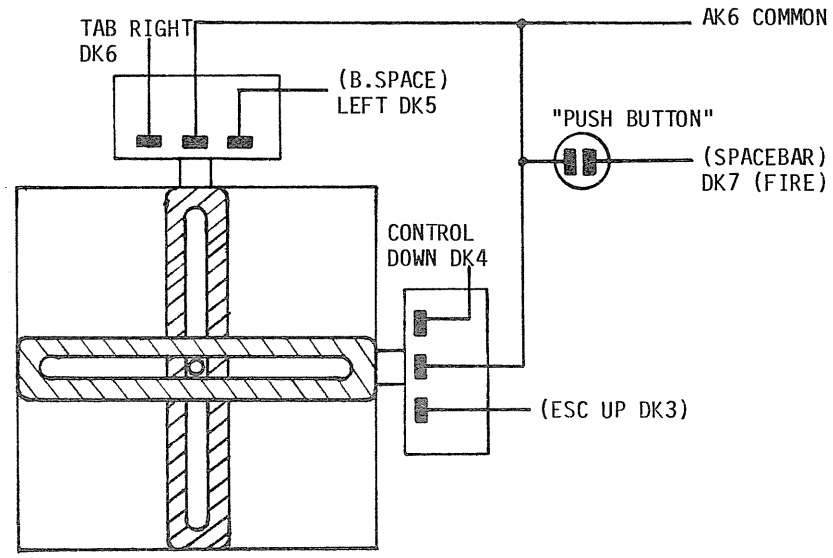
Purchase a joystick pot (5K), one push button switch, and a box to house them. Also needed is a length of 6 strand wire, and a 7 pin DIN plug and connector. Wire the keyboard first and mount the DIN socket on the front of the machine. Next, wire the joystick according to the diagram and, hey presto, you're in business. This method will work with all software which scans the arrow keys (which is just about the lot), or you can write your own software using the PEEK(14400) method.



Connect 2 centre terminals of joystick pots together.

I actually used a 4 channel mixer pot from DJ Reid Electronics ($4), and cut the carbon tracks inside to make it work. Test the joystick by using the following program:

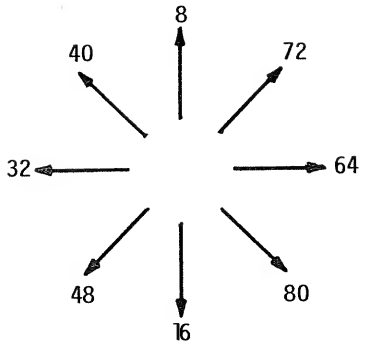
```
10 CLS:?"TESTING JOYSTICK"
20 A=PEEK(14400)
30 IF A=8 THEN?"UP";
40 IF A=16 THEN ?"DOWN";
50 IF A=32 THEN ?"LEFT";
60 IF A=64 THEN ?"RIGHT";
70 IF A=128 THEN ?"FIRE";
80 GOTO20
```



UNDERSIDE VIEW OF JOYSTICK

This will also allow multiple input, e.g. the eight directions as used in the game "ROBOT ATTACK" by BIGFIVE SOFTWARE (USA).

TOP VIEW
OF JOYSTOCK



Values (decimal) given to peek (14400) when joystick in these positions.

*N.B. Will not work with 100K pots (only 4 directors can be accessed. Use of 5K pots allows all 8 directions). Should any directions be reversed when using the joystick, swap the wires that are reversed on the joystick pots unitl it is working properly. I mounted the joystick in a small plastic box and used stick-on letters to indicate directions and fire button to smarten up the appearance. Altogether, it cost me $9.32(NZ). I never use the keyboard for playing games since "getting" the joystick, and it works just like my $50 one from the USA. Hope you enjoy it.

- 0000000000 -

**** SOFTWARE SECTION ****

***** CALENDAR - Peach and CC *****

With this program and a printer your colour computer can print a year calendar for any year this century in the range 1901-1999. All you need to enter is the year for which you want the calendar and the program does the rest.

Given the desired year, the day of the week of the first day of the year is calculated and a check is made for a leap year to adjust the number of days in February. Then a numeric array is filled in, in the same way one would do it manually if it was to be done by hand.

Before printing the calendar, a prompt is displayed to allow you to set the printer to the top of a page and then the data in the numeric array is converted to 80 character strings to be printed one line at a time. There are printer control codes issued to select double strike printing and these may need to be altered depending on the printer to be used. When the calendar is finished normal printing is restored and a form feed is sent to move to the top of the next page.

- 0000000000 -

***** HANGMAN - Tandy Colour Computer *****

This version of Hangman is loosely based on the program Super Hangman, originally published in the January '82 issue. The program provides graphics and sound at various stages of play.

Before each game you must press J or S to indicate Junior or Senior level. After this, the mystery word is chosen by the computer and the number of letters is shown at the bottom of the screen (each - represents a letter).

The game is played along the lines of the original HANGMAN games. You must guess the mystery word by guessing one letter at a time, or the whole word at once. For each letter guess, press the appropriate key. If the letter doesn't occur in the word, it will be displayed on the left of the screen and the GALLOWS will be added to. Six wrong guesses (letters and words) and you will lose. If you guess all the letters or the whole word then you will win. If you try a letter guess more than once, you will be told accordingly but not penalised. You will be warned when you are on your last guess.

To guess the whole word in one go, first press the @ key. You will then be told to type in your guess and press ENTER when you are finished. The computer will ignore all blanks (spaces) even if inside the word (i.e. DOG, D O G and DO G are all seen the same).

At the end of a game you can play again by pressing the Y key or end by answering with the N key. The program leaves very little space in a 16K machine but with more memory the vocabulary can be extended by adding more words to the list and changing the value of 418 in the line that chooses a word at random just before the DATA statements.

- 0000000000 -

***** SECTOR EDITOR - Peach - by Steve Payne *****

SECTEDT allows the following disk related operations:-

Display of a nominated disk sector
Modification of data within the nominated disk sector
Zeroing of a nominated range of disk sectors
Copying of a nominated range of disk sectors.

Hard copy of a displayed sector may also be requested.

DISPLAY/EDIT DISK SECTOR

COMMAND - 'D'(CR) (or simply (CR))

This is the default mode and may be selected by simply entering (Carriage Return) in response to the 'Function' prompt when the program is first run. In this mode the program will ask for:-

1) The drive number (from 0 to 3)
2) The track number (from 0 to 39)
3) The sector number (from 1 to 16)

The three parameters must be entered on the same line, separated by commas (',') and terminated by a (CR).

Please keep in mind that disk BASIC number DRIVES from 0 to 3, TRACKS from 0 to 39, but numbers SECTORS from 1 to 16.

Upon receipt of the (CR) SECTEDT will access the disk and display the sector contents across the top of the screen in 8 lines of the following format:-

1) The relative byte within the sector (left of the screen)
2) Sixteen two byte hex numbers representing the data
3) Sixteen ASCII characters whose codes are the hex numbers on the left of the screen. If the ASCII code falls between 20H and 7DH (32 to 125 decimal) then the actual ASCII character will be displayed. If the code is outside this range then it will be displayed as a decimal point ('.').

Following this display will be one line with the following data:-

Drive number,Track number,Sector number,Rel sector

These values are all in DECIMAL, the Rel sector being the total number of sectors on the disk up to the nominated one, (i.e. TRACK*16+SECTOR)

At this point the program will ask for another command. The options available to you are:-

1); = display next sector
2)- = display previous sector
3)R = re-display current sector
4)E = edit sector
5)Q = quit and return to main option list
6)J = enter new DRIVE/TRACK/SECTOR

All of the above commands are entered as a single letter (no (CR)) and the program will respond immediately.

DISPLAY SUB-COMMANDS

1) Display next sector, and
2) Display previous sector are self explanatory. They either increment or decrement the nominated sector number by one and then display the new sector's data.
3) Re-display current sector. This command will re-read the currently displayed sector from disk and then re-display it.
4) Edit sector. This command will put the program in edit mode. In this mode the program will display 'Edit' and wait for you to enter a two digit hex number in the range 00 to 7F. Upon receipt of a valid entry, the cursor will appear beneath the nominated byte. You may now enter a two digit hex number to replace the existing number or use one of the four arrow keys to move to another byte. When finished updating data in the sector, enter 'U' if you wish the updated sector to be written to disk, or 'Q' if you wish to return to display command mode.

Z- Zero all bytes from current cursor location to a nominated byte. The system will ask 'Zero bytes from XX to'. Enter a 2 digit hex number between the current byte (indicated by the cursor) to the end of the sector (7F). The sector will then be re-displayed with the nominated bytes zeroed.

S- Substitute characters from current cursor location to nominated byte. Same instructions as for ZERO BYTES, with the addition of a system request for a 'Fill character'. Enter any 2 digit hex number.

NB: BE CERTAIN THAT ANY ALTERATIONS YOU HAVE MADE TO THE SECTOR ARE CORRECT BEFORE ENTERING THE 'U' SUB-COMMAND - ONCE THE SECTOR HAS BEEN WRITTEN TO DISK IT IS TOO LATE TO CHANGE YOUR MIND!!

In addition to the above commands, hard copy of the displayed sector may be obtained by entering 'P' in either the display command mode or edit mode. Be aware, however, that any byte updates in edit mode MUST be made in groups of TWO hex numbers as the 'P' and arrow keys cannot be used when the cursor is beneath the second digit of the hex number.

5) Quit may be used in the display command mode to return control to the option list.
6) The 'J' command allows the selection of a different drive, track or sector without exiting and re-entering edit mode.

COPY DISK SECTORS

COMMAND - 'C' (CR)

This command is used to copy any number of sectors from one part of a disk to another or from one disk to another. The rules for DRIVE, TRACK and SECTOR numbers are the same as for entering display mode. The prompt for 'Sector count to transfer' means just what it says - the number of sectors you wish to copy. It must be a DECIMAL number between 1 and 16. Copies extending beyond the boundary of the nominated track are not supported.

The same warning applies here as in the 'U' command in sector edit mode. BE CAREFUL!! You can easily destroy a disk if you don't know what you are doing.

ZERO DISK SECTORS

COMMAND - 'Z' (CR)

This command is used to insert zeros (null characters) into every byte in the nominated sector(s). The same instructions apply here as for COPY DISK SECTORS.

MISCELLANEOUS

All the options excluding actual display of a disk sector involve disk write operations and, as such, can be disastrous if used without due care. If used sensibly, this program will give you the ability to directly edit or copy any desired area of the disk for patch application, etc. Be sure you have read and UNDERSTOOD the HITACHI Level 3 Disk BASIC Manual before embarking on a mad 'zapping' expedition, and pay particular attention to the section describing the disk map so that you will know which sectors are particularly sensitive.

As SECTEDT is written in BASIC, it is rather slow in some areas of operation, most noticeable being conversion and display of sector data. Please take this into account when running this program, and don't expect immediate screen updates. Also, while the program is in the process of displaying new sector data, DO NOT press any keys. The PEACH allows type ahead and if a character is entered while the processor is doing something else, you may well find that when it returns to command level, it has accepted the key you accidently (or otherwise) pressed and has gone off to 'do its own thing!!' With certain commands, you may not know what has happened.

- 0000000000 -

***** PROPERTY INVESTMENT SPECULATION LII/4K - by S. Mansell *****

This program arose from a need to determine in reverse how much I could afford to purchase, (instead of the other way around!). The beauty of this is that once you find how much you can borrow, you can add your deposit, put in the desired retail price and see what your repayments will be exactly. Then you can revise the figures over and over until you are happy with the right "mix" for yourself. As the program asks each question, the previous answer to that question is displayed to the left for your reference. This figure will, of course, be zero the first time the program is run.

The program prompts for the following information:

FORTNIGHTLY REPAYMENT - enter the amount you would like to repay each fortnight or enter zero if you wish to pay weekly.

WEEKLY RENTAL - enter the amount you would like to repay each week or if your answer to the first question was other than zero, then enter zero.

LOAN RATE - respond to this with the percent per annum, ie: 18%.

TERM OF REPAYMENT - type in the number of years that you would like the loan to be over. The program will then tell you what the maximum loan is that you can afford.

The program will then ask you what deposit you have. Answer this question and the program will display the maximum price that you can afford. After this, you then enter the desired purchase price. The program will then display information similar to that below, and also give you the option to print it out, after which the program will return to the start, displaying your last entries to the left, to give you the chance to adjust any or all of your answers.

SAMPLE OUTPUT

```
PROPERTY INVESTMENT SPECULATION
BY S.J. MANSELL  525-5665 1.2.81

FORTNIGHTLY REPAYMENT 45
WEEKLY RENTAL Ø
TOTAL MONTHLY REPAYMENT $ 90

LOAN RATE 15 %pa                    TERM 4 YRS
MAX. LOAN AFFORDABLE $ 2700
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
DEPOSIT 300
MAX. PRICE AFFORDABLE $ 3000
-------------------------
DESIRED PRICE          $ 3000
     MONTHLY PAYMENTS $ 90

                - 0000000000 -
```

***** FASTER CASSETTE ROUTINES LII/16K m/l - by J.S. Langsford *****

HOW TO LOAD FASTER

No memory has to be reserved to load this program. It reserves its own. This allows reserving of memory for other machine code routines. The only stipulations are that the end of BASIC Pointer should not be above 7ØØØH (28672) and that memory already reserved should not be below 75D3H (3Ø563). This leaves just over 2K on 16K machines and bulk more on 32K or 48K machines for other machine code routines. If reserved memory is already below this point an OM Error will result. If the end of BASIC Pointer is above 7ØØØH it will still load and work but any BASIC program in memory will be destroyed. In any case, BASIC variables will be reset to zero for numerical data and nulls for string data. This is because the string space is interfered with during setting up. If both of the above conditions are met, any BASIC program in memory will be left untouched. This does not leave much room for BASIC Programs to be left untouched on disk based systems but leaves plenty on a non-disk system. (By loading it here each type of user has advantages and limitations).

To load and initialize the program, simply load it using the SYSTEM command with a file name of FASTER. When loaded, type a single /. Memory will then be reserved, the program adjusted to its new location and shifted there. In some cases, it may be necessary to reinitialize the program so the screen is cleared and the new entry point printed. The program then jumps back to BASIC's
READY
>-
message to await use.

The version of FASTER supplied on the distribution disk has been given an ORG of 8ØØØH to avoid any conflict with the different DOS's that may be used. If you are typing the program in from the magazine it is recommended that you use this as an ORG and you will need EDTASM-PLUS to assemble the source code.

TRSDOS or Distribution DOS.
From DOS type:

```
LOAD FASTER/CMD   (ENTER/NEWLINE)
BASIC             (ENTER/NEWLINE)
SYSTEM            (ENTER/NEWLINE)
/32768            (ENTER/NEWLINE)
```

An initialization message will appear at the top of the screen with the program's entry point.

NEWDOS 80 V1.Ø or V2.Ø DOSPLUS
Load using the above method or from BASIC simply type:

```
CMD"FASTER  (ENTER/NEWLINE)
```

The program will initialize as above.

NOTE: Disk users must use CMD"T" to disable interrupts before doing any cassette I/O.

MICRO-80 PRODUCTS CATALOGUE

This catalogue contains a selection from the wide range of peripherals, interfaces, computers and software carried by MICRO-80 for your computer. If you don't see the item you want, contact us, we probably have it anyway!

MICRO-80 has been supplying customers throughout Australia and the Pacific region by mail-order for 2½ years. Our customers find this a simple and efficient way to do business. You may place your order by telephone or by mailing the order form from any issue of MICRO-80 magazine. Generally, it takes about one week from receipt of order until despatch. You should allow 2-3 days for your letter to reach us and 7-10 days for the parcel to reach you, making a total turnaround time of 2½-3 weeks.

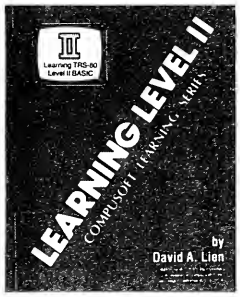
WARRANTY AND SERVICE

All hardware products carry a 90 day parts and labour warranty either from the manufacturer/distributor or from MICRO-80 Pty Ltd. In many cases, warranty servicing can be arranged in your own city, otherwise goods will be repaired by our own team of technicians in our Adelaide workshops.

TRADE-INS AND TERMS

MICRO-80 can accept your existing equipment as a trade-in on new equipment. We can also arrange consumer mortgage financing or leasing on larger hardware purchases. Contact us for details.

BOOKS



LEARNING LEVEL II
by David A. Lien

Written by the author of the Level I Users Manual, *Learning Level II* covers all Level II BASIC beyond Level I, plus much more. It shows you how to use the Editor, explains what the many error messages are really saying, and leads you through conversions of Level I programs to Level II.

Dual cassettes, printers, the Expansion Interface with clock and other features are explained in the same easy-to-learn style that made the Level I Manual famous. *Learning Level II* is an invaluable supplement to the TRS-80 Level II and System 80 manuals and is now only $7.95 (plus $1.20 p&p).

BASIC FASTER AND BETTER AND OTHER MYSTERIES
by Lewis Rosenfelder

Basic is not nearly as slow as most programmers think. *Basic Faster and Better* shows you how to super charge your BASIC with almost 300 pages of fast, functions and subroutines.

You won't find any trivial poorly designed "check-book balancing" programs in this book — it's packed with *useful* programs.

Tutorial for the beginner, instructive for the advanced, and invaluable for the professional, this book doesn't just talk . . . it shows how! Basic Faster and Better is $32.50 (plus $1.20 p&p).

BOOKS

THE CUSTOM TRS-80 AND OTHER MYSTERIES
by Dennis Bathory Kitsz

Ever wanted to do things to your TRS-80 that Radio Shack said couldn't be done? How about reverse video, high resolution graphics, and audible keystrokes?

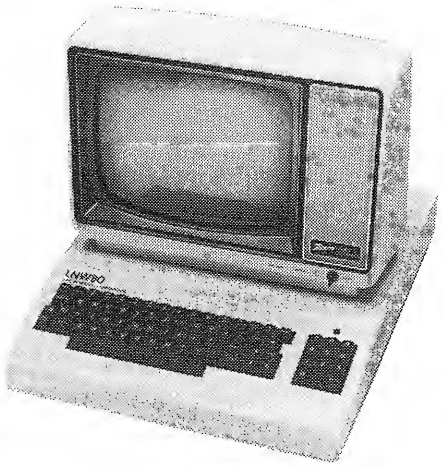
Now enough? How about turning an 8-track into a mass storage device, making music, controlling a synthesiser, individual reverse characters, and a real-time clock just to name a few?

The *Custom TRS-80 and Other Mysteries* is packed with more than 290 pages of practical information and can be yours for only $32.50 (plus $1.20 p&p).

TRS-80 DISK AND OTHER MYSTERIES
by H.C. Pennington

TRS-80 Disk and Other Mysteries is the definitive fix-it book for disk users. More than 130 pages of easy to read, entertaining and immensely useful information. Find out how to recover disk files, the layout of information on disks, memory maps, problem solutions . . . the list goes on! Many readers have saved days of work by recreating disk files that were unreadable. *TRS-80 Disk and Other Mysteries,* which has received favorable reviews in several magazines, is yours for only $27.00 (plus $1.20 p&p).

THE LNW80 MkII MICROCOMPUTER



Manufactured in America by LNW Research Corporation, the LNW80 II has the following outstanding features:

- Completely software and hardware COMPATIBLE with the TRS-80 Model 1.
- HIGH RESOLUTION COLOUR GRAPHICS — 4 MODES:
  - B/W LO-RES 128 x 48
  - B/W HI-RES 480 x 192
  - COLOUR LO-RES 128 x 192 in 8 COLOURS
  - COLOUR HI-RES 480 x 192 in 8 COLOURS
- CP/M Disk Operating System.
- Single and Double Density Disk Operation.
- Supports 5¼ inch or 8 inch Floppy Disk Drives.
- 48K RAM in TRS-80 mode plus 16K High Resolution graphics RAM.
- 64K RAM in CP/M mode plus 32K Banked in, usable in BASIC, plus the 16K High Resolution Graphics RAM.
- 4 MHz Z80A microprocessor — over twice the operating speed of the Model 1.
- HI-RES COLOUR (R-G-B) and B&W video outputs.
- 3 screen display modes:
  - 64 characters x 16 lines
  - 80 characters x 16 lines
  - 80 characters x 24 lines
- SOFTWARE SUPPORT

Apart from being able to run all TRS-80 Model 1 software and all CP/M software, there is also an extended BASIC interpreter available for the LNW80 II using most of the same commands as the TRS-80 Colour Computer but with full LNW Graphics Resolution, SET, RESET, POINT, LINE and CIRCLE as well as special commands to generate sound effects and tones. TRS-80 Colour Computer BASIC programs can be transferred to the LNW with only minor changes.

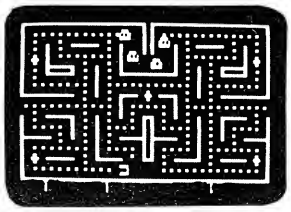
Prices include Sales Tax and are subject to change without notice. Prices are FOB Adelaide. Add $20 road freight anywhere in Australia.
All equipment carries MICRO-80's Australia-wide 90-day warranty covering parts and labour.

The LNW80 II is the ideal computer for the serious hobbyist or businessman who is seeking a higher performance, more reliable computer to replace his TRS-80 Model 1 without sacrificing his investment in software or his programming experience. The LNW80 II uses standard Tandy or Tandy compatible disk drives. If you already have a disk TRS-80 system you may continue to use your existing disk drives on the LNW80 II.

LNW80 II Computer — complete except for disk drives and monitor Includes:

— CP/M Disk Operating System Dosplus 3.4 Double Density Disk Operating System
— LNW Extended Colour Basic Interpreter ..... **$2750** INC.S.T.

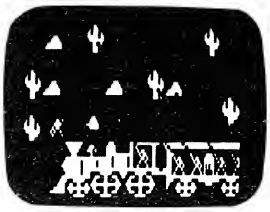
HI-RES Green Phosphor Monitor ..... **$265** INC.S.T.
Super HI-RES Hitachi RGB Colour Monitor ..... **$1250** INC.S.T.
Two Singlesided 40 Track Double Density Disk in cabinet with power supply and cable ..... **$825** INC.S.T.



SCARFMAN

This incredibly popular game craze now runs on your TRS-80! It's eat or be eaten. You run Scrarfman around the maze, gobbling up everything in your path. Try to eat it all before nasty monsters devour you. Excellent high speed machine language action game from the Cornsoft Group. With sound.

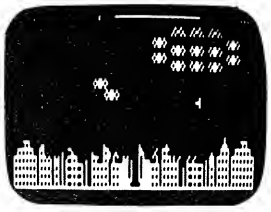
Price: $17.95



THE WILD WEST

It's up to you to keep the West beautiful with Outlaws and renegade Indians on all sides. Even the train has been captured by Outlaws with all the payroll on board. Can you clean up the Wild West?

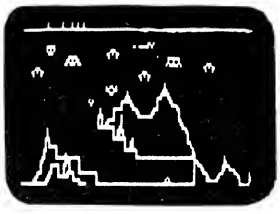
Price: $26.50



SPACE ATTACK

Steady your nerves, keep a sharp lookout, and prepare for battle to save your city. Fiendish aliens are all around, and if they destroy the city you lose.

Price: $26.50



STRIKE FORCE

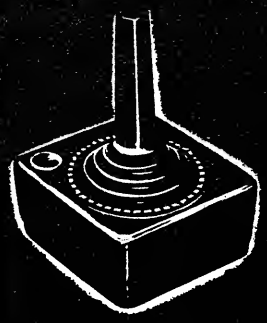
As the primary defender of a world of cities under deadly alien attack, your weaponry is the latest rapid fire missiles, long range radar, and incendiary "star shells." Your force field can absorb only a limited number of impacts. A complex game of strategy, skill and reflexes from Melbourne House.

Price: $26.50

PENETRATOR

Soar swiftly over jagged landscape, swooping high and low to avoid obstacles and enemy missiles attacks. With miles of wild terrain and tunnels to penetrate, you're well armed with bombs and multiple forward missile capability. From Melbourne House. Features sound, trainer mode and customizing program.

Price: $36.50



STICKEROO JOYSTICK INTERFACE

for the TRS-80 MODELS I & III* and SYSTEM 80

FROM **$32.00**

ADD $2.00 p. & p.

CONVERT YOUR COMPUTER INTO AN ARCADE GAMES MACHINE

Micro-80's Stickeroo Interface Features:

•Compatible with Joysticks for Atari, Vic-20 and most video games •Saves your keyboard from abuse •Compatible with programs from leading US software houses: Big Five, Cornsoft, Melbourne House, Adventure International •Adds a whole new dimension of pleasure and fun to your favourite games •Will be supported in MICRO-80 •Can be used with your own basic or ML Programs •Comes complete, ready to plug in and use •Absolutely no modifications required to your computer

Due to popular demand, Stickeroo Interface is now available separately so you can use the Joystick of your choice.

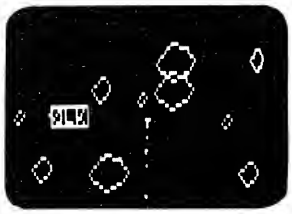
PRICE INCLUDES ... STICKEROO + INSTRUCTIONS + DEMO PROGRAM LISTING
PLEASE SPECIFY TRS-80 MODEL I or III* OR SYSTEM 80 WHEN ORDERING

*Model III Interface available soon at slightly higher price, to be announced.

PISTOL GRIP JOYSTICK WITH FIRE BUTTON

$25 + $2 p & p (No p & p required if ordered with Stickeroo Interface)

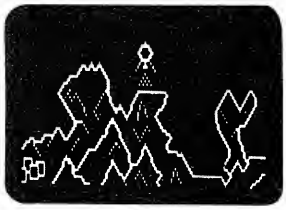
ALL GAMES ADVERTISED ON THIS PAGE ARE STICKEROO COMPATIBLE



SUPER NOVA

Asteroids float ominously around the screen. You must destroy the asteroids before they destroy you! (Big asteroids break into little ones). Your ship will respond to thrust, rotate, hyperspace and fire. Watch out for that saucer with the laser! As reviewed in May 1981 Byte Magazine.

Price: $26.50



LUNAR LANDER

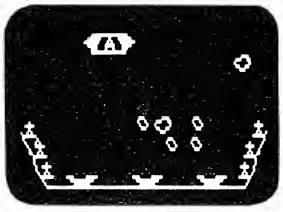
As a vast panoramic moonscape scrolls by, select one of many landing sights. The more perilous the spot, the more points scored -- if you land safely. You control LEM main engines and side thrusters. One of the best uses of TRS-80 graphics we have ever seen. From Adventure International. With sound.

Price: $26.50

COSMIC FIGHTER®

Your ship comes out of hyperspace under a convoy of aliens. You destroy every one. But another set appears. These seem more intelligent. You eliminate them, too. Your fuel supply is diminishing. You must destroy two more sets before you can dock. The space station is now on your scanner. With sound!

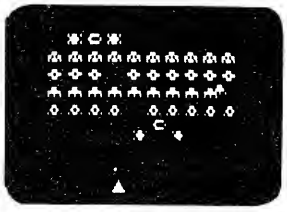
Price: $26.50



METEOR MISSION II

As you look down on your view, astronauts cry out for rescue. You must maneuver through the asteroids and meteors. (Can you get back to the space station?) Fire lasers to destroy the asteroids, but watch out, there could be an alien Flagship lurking. Includes sound effects!

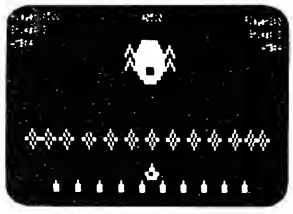
Price: $26.50



GALAXY INVASION

The sound of the klaxon is calling you! Invaders have been spotted warping toward Earth. You shift right and left as you fire your lasers. A few break formation and fly straight at you! You place your finger on the fire button knowing that this shot must connect! With sound effects!

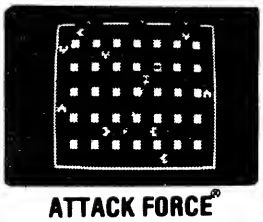
Price: $26.50



DEFENSE COMMAND

The invaders are back! Alone, you defend the all important nuclear fuel canisters from the repeated attacks of thieving aliens, repeatedly. An alien passes your guard, snatches a canister and flys straight off. Quick! You have one last chance to blast him from the sky! With sound and voice.

Price: $26.50



ATTACK FORCE®

As your ship appears on the bottom of the maze, eight alien ships appear on the top, all traveling directly at you! You move toward them and fire missiles. But the more aliens you destroy, the faster the remaining ones become. If you get too good you must endure the "Flagship". With sound effects!

Price: $26.50

FOR YOUR ENTERTAINMENT

MICRO-80 now offers you the widest range possible in entertainment software. These programs are supplied on cassette for the Level II/16K TRS-80 Model I/III (except as noted). They are also suitable for the System 80 but sound may not be available unless a hardware modification has been fitted to reverse the roles of recorders #1 and #2. *Order yours now while stocks last!*

ELIMINATOR **$26.50**
Your mission is to prevent the marauding alien hoards from recovering your energizers from the planet surface. There are several types of alien ships — each with different weapons to destroy you!! — with sound effects.

PLANETOIDS **$26.50**
It's your ship against a swarm of killer planetoids, as you try to destroy them before they destroy you — with sharp graphics and sound effects.

STAR FLITE: Voyage of the Enterprise **$26.50**
You are the commander of the Star Ship Enterprise and your mission is to explore the, as yet, unknown Omega VI region of the galaxy. But, beware! The Klingons have recently invaded this region and you must put a stop to it. Good luck!

ADVENTURE **$42.00**
This is the original ADVENTURE game. The game fills an entire diskette and provides endless variety and challenge as you seek to rise to the level of Grand Master. Until you gain skill, there are whole areas of the cave that you cannot enter. (Requires 32K, one disk. For Model I/System 80 only.)

MISSILE ATTACK **$26.50**
This is a real-time game with sound effects. You must protect your cities against enemy missiles, as your skill increases, so does the level of difficulty making accuracy a must.

SPACE INTRUDERS **$26.50**
A very fast game with the deluxe version of Space Invaders, complete with "spitting" invaders and the SOS of escaping aliens — with sound effects.

SARGON II **$42.00**
One of the best micro-computer Chess programs, SARGON II has 7 levels of play (level 0-3 in tournament time), allows you to set up the board and a special hint mode is available (except level 0) to suggest a good (but not necessarily the best) move for you when you need help.

OLYMPIC DECATHLON **$56.00**
This exciting skill game takes you through all 10 events from the pole vault to the 100-metre dash. Each event is presented with animated graphics that put you directly into the action. To win the gold medal, you'll need a good sense of timing, fast reflexes and good co-ordination. (For Model I/System 80 only.)

PLUS
NEW SOFTWARE
From Cosmic Software and Fantastic Software:

Star Cresta	$20.95
Defence Penetrator	$20.95
Stellar Warp	$20.95
Outland	$25.95
Bosk One	$25.50
Panik	$28.50
Rally Racer	$20.95
Morgoth	$20.95
Devil's Tower	$25.95
Doomsday Mission	$25.50
Desert Peril	$28.50
Alien Taxi	$28.50

NOTE:
As the prices of imported software may vary, these prices are valid for current stock only and prices are subject to change without notice.

BUY YOUR MODEL 3 FROM MICRO-80 AND SAVE $00's

MICRO-80 fits reliable MPI disk drives to the TRS-80 Model 3 to give system capacities and capabilities far in excess of those available elsewhere. All our conversions utilise low dissipation, switching-mode supplies to avoid screen jitter and overheating. The disk controller boards used incorporate special compensation circuitary for 80 track disk drives and may also be used to run 8 inch disk drives with an appropriate cable and DOS.

MODEL 340 **$3130**

2 40 TRACK SINGLE-HEAD DISK DRIVES GIVING
350K FORMATTED STORAGE, 48K RAM

MODEL 340+ **$3350**

2 40 TRACK DUAL-HEAD DRIVES GIVING
700K FORMATTED STORAGE, 48K RAM

MODEL 500 — 5+ MEGABYTE MODEL 3 **$5895**

1 40 TRACK DUAL-HEAD DRIVE GIVING 350K
OF FLOPPY DISK STORAGE FOR TRANSFERRING
PROGRAMS AND BACKUP, 48K RAM, EXTERNAL
5 MEGABYTE WINCHESTER SUB-SYSTEM,
DOSPLUS 4.0 DISK OPERATING SYSTEM
The MODEL 500 offers the high speed, mass storage capacity and reliability of a Winchester drive for thousands of dollars less than you would pay for any comparable system. Model 500 is a serious business computer able to tackle the most demanding tasks.

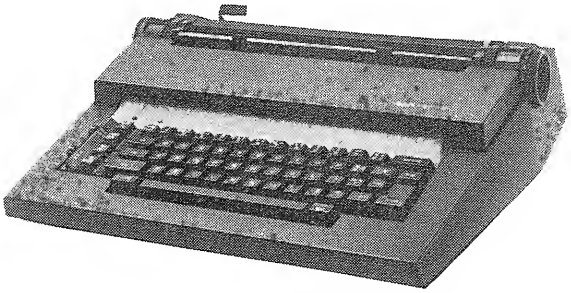
WINCHESTER DISK DRIVE SUB-SYSTEM **5MByte $2995**
10MByte $3750

This Winchester Disk Drive sub-system provides either 5 or 10 Megabyte of reliable, high speed storage. It connects to any standard Model 3 equipped with one or more floppy disk drives and does not void the Tandy warranty. Complete with DOSPLUS 4.0 Disk Operating system.

Prices include Sales Tax and are subject to change without notice. Prices are FOB Adelaide. Add $20 road freight anywhere in Australia. All computers and peripherals carry MICRO-80's Australia-wide. 90-day warranty covering parts and labour.

Daisy Wheel Printers/Typewriters

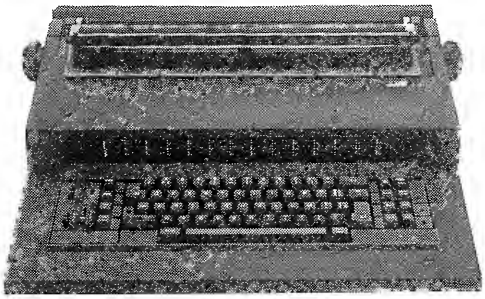
OLIVETTI PRAXIS 35



$895

plus $10 road freight anywhere in Australia

OLIVETTI ET-121



$1500

plus $20 road freight anywhere in Australia

MICRO-80 has converted these OLIVETTI typewriters to work with the TRS-80, SYSTEM 80 or any other microcomputer with a Centronics parallel port. Now you can have the best of both worlds — an attractive, modern, correcting electronic typewriter which doubles as a correspondence quality Daisy Wheel printer when used with your microcomputer.

The **PRAXIS** is a portable typewriter, designed for private and light commercial use with an average print speed of 12 c.p.s.

The **ET-121** is a large typewriter intended for heavier duty and features a print speed of up to 17 c.p.s.

Centronics printer cable to suit TRS-80 or SYSTEM 80 . . . . . $39

MICRO-80 is an A-Grade Olivetti distributor and has been producing printer conversions for Olivetti daisy wheel typewriters for over 12 months. Write or call for full details.

16K Memory Upgrade Kit

$30

plus $2.00 p. & p.

Large volume means we can buy better and can pass the savings on to you. There are our proven, prime, branded 200 manosecond chips, guaranteed for 12 months. A pair of DIP shunts is also required to upgrade CPU memory in the TRS-80 — these cost an additional $4.00. All kits come complete with full, step-by-step instructions which include labelled photographs. No soldering is required. You do not have to be an experienced electronic technician to install them.

Lower Case Modification

$49

plus $2.00 p. & p.

The MICRO-80 modification features true below-the-line descenders, a block cursor and symbols for the 4 playing-card suits. Each kit comes with comprehensive fitting instructions and two universal lower-case driver routines on cassette to enable you to display lower case. These routines are self-relocating, self-protecting and will co-reside with other machine language programs (the second includes keyboard-debounce and flashing cursor). Fitting requires soldering inside the computer and should only be carried out by an experienced hobbyist or technician. A fitting service is available in capital cities for only $20.00 and a list of installers is included with each kit. (Specify TRS-80 Model I or System 80 when ordering.)

All prices include Sales Tax and are correct at time of publication but are subject to changes without notice. All equipment carries MICRO-80's Australia-wide 90-day warranty covering parts and labour.

DISK OPERATING SYSTEMS & DEVELOPMENT SOFTWARE

You can increase your programming productivity, the execution speed and 'user friendliness' of your programs by using an enhanced Disk Operating System (DOS). Together with the other utility software, you can get the most from your disk drives.

DOSPLUS 3.3 **$99.95**
(Specify Model I single or double density or Model III)
An economic DOS intended for the first-time user and requiring single-sided disk drives. (The TRSDOS & DISK BASIC MANUAL is required to supplement the DOSPLUS manual).

DOSPLUS 3.4 **$149.95**
(Specify Model I single or double density or Model III)
With a high degree of compatibility with TRSDOS, DOSPLUS 3.4 supports single- or double-sided, single or double density, 5″ or 8″ disk drives with any track count (up to 96). Suitable for the first-time or experienced user wanting a fuss-free, bug-free, easy to understand but very powerful DOS which supports variable length records up to 255 bytes long. Comes with a stand alone manual.

ENHBAS **$52.95**
ENHBAS adds over 30 new commands and functions to your BASIC interpreter including high speed SORT, labels in BASIC, RESTORE to any line number, WHILE-WEND for structured programming, SCROLL, LEFT, INVERT, DRAW and PLOT to give you ease of control over graphics, SOUND and PLAY to add realistic sound effects and many more. Makes programming a breeze! Available for Model I or III, disk or cassette — specify which when ordering.

NEWDOS 80 VERSION 2.0 **$169.00**
(Specify Model I or Model III)
Newdos 80 suits the experienced user who has already used TRSDOS, understands the manual and is prepared to learn the somewhat complicated syntax of one of the most powerful DOS's available. With the correct hardware, Newdos 80 supports any mix of single- or double-sided, single or double density, 5″ or 8″ disk drives with track counts up to 96. It provides powerful, flexible file handling in BASIC including variable length records up to 4096 bytes. Definitely not for the beginner.

MASTER DISK DIRECTORY **$20.95**
FIND THE PROGRAM FAST!! PAYS FOR ITSELF BY RELEASING REDUDANT DISK SPACE!! MASTER DIRECTORY records the directories of all your individual disks onto one directory disk. Then it allows you examine them, find an individual file quickly, list files alphabetically, weed out redundant files, identify disks with free space, list files by extension, etc., etc. This program is invaluable for the serious disk user and will pay for itself many times over.

THE FLOPPY DOCTOR/MEMORY DIAGNOSTIC
Model I Disk **$36.50** Model III Disk **$43.50**
THE MICRO CLINIC offers two programs designed to thoroughly check out the two most trouble-prone sections of the TRS-80 — the disk system (controller and drives) and the memory arrays. Both programs are written in Z80 machine code and are supplied together on diskette for a minimum 32K, one disk system. Specify Model I or Model III.

MORE ENTERTAINMENT SOFTWARE

ADVENTURE HINT BOOK **$10.95**
If you can not go any further this will give you clues that may help — written by Scott Adams for Adventures 1—9.

LABYRINTH **$26.50**
Labyrinth — you move through a gigantic labyrinth and scattered through this nightmare are a multitude of objects and obstacles. A minotaur prowls the corridors — you must kill it before it kills you, Labyrinth has over 550 locations — be patient.

ASYLUM **$26.50**
Asylum places you in a cell, you have to escape. It's harder than it sounds, lots of hazards will be encountered.

DEATHMAZE 5000 **$26.50**
Deathmaze 5000 is another 3-D adventure. You move through a 5 storey building — your goal is to leave the deathmaze alive.

Here at Last!
The
EPSON USER'S MANUAL
by David A. Lien

MX-80 USER'S MANUAL ..... $16.95 plus $1.00 p&p

MX-100 USER'S MANUAL ..... $16.95 plus $1.00 p&p

MICRO-80 HELPS YOU BEAT DEVALUATION!

The recent devaluation of the Australian dollar has increased the prices of imported printers. MICRO-80 helps you beat the price rises by offering a 50% discount on all prices.

EPSON MX80 III
80c.p.s., 80 col., bit graphics, dot matrix.

$999 LIST PRICE $1050

EPSON MX100 III
80 c.p.s., 132 col., bit graphics, dot matrix.

$1500 LIST PRICE $1580

ITOH PROWRITER 8510
120c.p.s., 80 col., bit graphics, proportional print dot matrix.

$1150 LIST PRICE $1370

ITOH PROWRITER 1550
120c.p.s., 132 col., bit graphics, proportional print dot matrix.

$1299 LIST PRICE $1244

MICROLINE 84
200c.p.s., 132 col., bit graphics, dot matrix.

$2110 LIST PRICE $2289

ITOH F10-40P
40c.p.s., 132 col., correspondence quality daisywheel.

$2535 LIST PRICE $2880

All prices include Sales Tax and are correct at time of publication but are subject to change without notice. All equipment carries MICRO-80's Australia-wide 90-day warranty covering parts and labour.

HOW TO USE FASTER

Once initialized, this program adds several new statements to BASIC. All the usual BASIC statements will work as before. At this stage no DOS or disk BASIC exits are used so it will not interfere with disk BASIC or DOS operation. The new operations are obtained by preceding several normal BASIC statements with a '#' sign. The new operations work in a similar fashion to the original statements but differ slightly so I will describe each individually.

#CLOAD works exactly the same as CLOAD but at 1ØØØ baud. Of course, it is not possible to read tapes written at 5ØØ baud using the 1ØØØ baud statement, and visa-versa. #CLOAD? will verify a 1ØØØ baud program just as CLOAD? will for a 5ØØ baud tape. #CLOAD#-1,"N" will load a tape with the name N at 1ØØØ baud just as CLOAD#-1,"N" will at 5ØØ baud. The equivalent statement #CLOAD"N" will also work.

#CSAVE works at 1ØØØ baud in every way the same as normal CSAVE does at 5ØØ except that a number of copies may be specified up to 255. #CSAVE1Ø"N" will write 10 copies of the program in memory with the name N. The default value if no number is specified is 1. The following are all equivalent statements:

```
#CSAVE1"N"
#CSAVE"N"
#CSAVE#-1,"N"
#CSAVE1#-1,"N"
```

#PRINT will write data to tape at 1ØØØ baud. To write data to tape at 5ØØ baud a cassette number must be specified but with this routine it is not essential but may be included. The default cassette number is cassette 1.

#INPUT will input data from cassette written at 1ØØØ baud. As with #PRINT no cassette need be specified but, again, it may be if required. As with INPUT, #INPUT may not be used as a direct statement.

#LIST is an extra feature which can be used to update programs to the new 1ØØØ baud routines. Its function is to list the BASIC statements in the program in memory which contain cassette operations. This includes the new ones. As with all other added functions except INPUT or #INPUT, it may be used in a program or as a direct command.

#ON is another added feature. Its function is to turn on the motor of the specified cassette until any key is pressed. The default cassette, if none is specified, is 1. This may be useful for rewinding tapes. It may also be useful in a program which prompts the user. If a cassette is specified it must be followed by a comma. The format of the statement is #ON#-1, or just #ON.

HOW THE DATA RATE IS INCREASED

When data is written to cassette, each data pulse is preceded by a synchronization pulse. Because the Z-80 is an 8 bit machine, 8 bits are written at a time. This means each byte written requires the equivalent time of 16 pulses to write 8 bits. With the new routines a sync pulse is written every 8 bits. This is equivalent in time to writing only 9 pulses for every 8 bits. I am not pretending that it is as reliable as the 16 pulse method, but I have not yet had a loading error using these routines, and since writing them, I have stored hundreds of megabytes onto tape using them. This method multiplies the baud rate by 16/9 which almost doubles it. (BAUD stands for bits per second, by the way). A few extra savings in time are achieved along the way, but the same delay between resetting and reading the read flip-flop is used, so if XRX modifications work with normal cassette operations, they will also work with these routines. For more information look at the source file for the program.

HOW THIS PROGRAM INTERCEPTS BASIC OPERATION

There is no way to change what is in ROM unless you burn it out or hit it with a hammer. That is, of course, unless it is in EPROM. Adding statements to BASIC must be done from RAM. All other programs I have seen which attempt to change the way BASIC functions have done so by using DOS, or disk BASIC exits which have been provided in the ROM. These are calls to RAM addresses which are initialized to returns or jumps to an ?SN Error in non-disk systems. This program does neither. One DOS exit is used temporarily but is returned to its original state when its use is no longer required.

The method I have used is a little more complex. There is in ROM another type of call to RAM. These are RST instructions. They are similar to subroutine calls except they only require 1 byte instead of 3. There is one of these at the start of statement execution. The difficulty with using it is that it is called 56 times in ROM and probably by other programs in RAM such as DOS. The solution is to use it anyway and make a test at the start to see where the RST 1ØH is being called from. This is easy because there is a return address on the stack to test. This will point to the next instruction after the RST 1ØH call. If the first value on the stack is not the one we are looking for, execution is passed back to the original jump taken by the RST 1ØH call (JP 1D78H). If it is the right one, control is taken over by FASTER.

ONE FINAL NOTE FOR DISK USERS

Pressing the reset button or booting the DOS after this program has been initialized can stop its operation, as the RST vectors are reset in these circumstances. This is no great problem. All that is required is to reinitialize it. This is done by typing system and then a / followed by the entry point given when the program was first loaded. It may be a good idea to write it down somewhere if you expect one of the above situations to occur.

- 0000000000 -

**** CRICKET LII/16K - by G. Bull *****

Cricket was the first game I wrote and my recollections of the finer points of the program have been dulled by time. The main working functions are as follows:

1. The program makes extensive use of the RND function to set -

   (a) the origin of the ball,
   (b) the flight of the ball prior to hitting the pitch,
   (c) the angle of deflection upon hitting the pitch,
   (d) the speed of the ball,
   (e) the number of runs scored.

2. All parameters relating to ball speed and angles are derived prior to the ball being bowled.

3. Ball speed is derived by varying the distance between each "SET" and corresponding "RESET". As the time taken to perform the SET and RESET functions is constant, the greater the distance between consecutive SETS, the faster the ball speed.

4. The POINT function is used to detect -

   (a) the ball hit by bat,
   (b) the ball hitting the stumps.

   It is possible for the ball to go between the stumps.

5. The Field: There are two fixed fields, being generated by data statements. The close in field operates exclusively for the first 20 balls or so, then alternates with a further field setting. A catch is detected when the ball SET position equals the "Field" set position. Movement is continued until screen limits are reached to allow the player time to see the catch. It is possible for a ball to go above a fielder.

6. Bat positioning has the non-orthodox approach of one Key Press equals one position moved for reasons of -

   (a) maintaining a more constant ball speed, and
   (b) batsmen usually protect their wicket rather than chase a ball as tennis or squash players would.

7. Bowlers tire by increasing the timing loop for a SET prior to RESET being executed.

This game offers four different levels of play, the fastest being one day limited over (50 balls), or Test cricket (100 balls) and the slowest, English cricket (300 balls).

An innings is over at the fall of the tenth wicket. You are the batsman in this game. To move your bat up, press the "/" key and to move the bat down, press the "Z" key. The bat will move one position for each press.

- 0000000000 -

***** POTHOLE LII/16K - by S. Crocket *****

You have one pothole cover for four potholes. Points are scored each time you stop a character falling down the hole, one point for each character on the screen.

The game finishes if three characters fall down a hole. If you score over 250 an extra character can fall before the game is over.

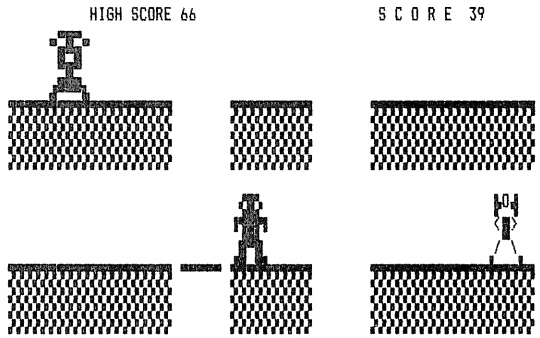
To position the pothole cover, use the following keys:

↑(ESC)	Up	
↓(CTRL)	Down	
<	Left	(unshifted)
>	Right	(unshifted)

The game is started by selecting a skill level between 0-9 where 0 is fast and 9 is slow. The program takes about 8 seconds to initialize.

- 0000000000 -

VIEW OF SCREEN FROM POTHOLE



**** CALENDAR ****
COLOUR COMPUTER

```
10 '**** CALENDAR ****
20 'WRITTEN BY R.J.WIWATOWSKI
30 'MODIFIED FOR THE TRS80CC
40 CLEAR600
60 DIML(11),YR(11,6,4),X$(9,9)
70 DATA31,28,31,30,31,30,31,31,3
0,31,30,31
80 DATA" 000000 ","00000000","00
    00","00    00","00    00","0
0    00","00    00","00    00","
00000000"," 000000 "
90 DATA"    1   ","   11   ","
111   ","   11   ","   11   ","
  11   ","   11   ","   11   ","
 111111 "," 111111 "
100 DATA" 222222 ","22222222","2
2    22","       22","  222222","
 222222 ","222     ","22      ",
"22222222","22222222"
110 DATA"33333333","33333333","
    33 ","    33  ","   333  ","
    333 ","     333","33    33",
"33333333"," 333333 "
120 DATA"     4  ","    44  ","
  444  ","  4444  "," 44 44  ","
44  44  ","44444444","44444444",
"    44  ","    44  "
130 DATA"55555555","55555555","5
5      ","55      ","5555555 ","
55555555","      55","55    55",
"55555555"," 555555 "
140 DATA" 666666 ","66666666","6
6    66","66      ","6666666 ","
66666666","66    66","66    66",
"66666666"," 666666 "
150 DATA"77777777","77777777","
     77","      77","     77 ","
    77  "," 77   ","   77   ","
"   77   "," 77   "
160 DATA" 888888 ","88888888","8
8    88","88    88"," 888888 ","
 888888 ","88    88","88    88",
"88888888"," 888888 "
170 DATA" 999999 ","99999999","9
9    99","99    99","99999999","
 9999999","      99","99    99",
"99999999"," 999999 "
180 FORI=0TO11:READL(I):NEXTI
190 FORI=0TO9:FORJ=0TO9:READX$(I
,J):NEXTJ:NEXTI
200 PRINT#-2,CHR$(27)+"!";:'SET
ITOH EMPHASISED MODE ("G" FOR EP
SON)
210 GOTO550
220 FORM=0TO11
230 W=0:DT=1
240 YR(M,D,W)=DT:DT=DT+1:D=D+1
250 IFD>6THEND=0:W=W+1:IFW>4THEN
W=0
260 IFDT<L(M)+1THEN240
270 NEXTM
280 RETURN
290 Q$= RIGHT$(STR$(Y),4):I1= VA
L(MID$(Q$,1,1)):I2= VAL(MID$(Q$,
2,1)):I3= VAL(MID$(Q$,3,1)):I4=
VAL(MID$(Q$,4,1))
300 PRINT#-2:PRINT#-2:PRINT#-2
310 PRINT#-2,TAB(20)"M   M III
CCC  RRRR  OOO       88   00"
320 PRINT#-2,TAB(20)"MM MM  I  C
   C R  R O   O     8  8 0  0"
330 PRINT#-2,TAB(20)"M M M  I  C
    RRR  O   O ***  88  0  0"
340 PRINT#-2,TAB(20)"M   M  I  C
   C R  R O   O     8  8 0  0"
350 PRINT#-2,TAB(20)"M   M III
CCC  R  R  OOO       88   00"
360 PRINT#-2:PRINT#-2
370 PRINT#-2,"     "+ STRING$(70
,"*"):PRINT#-2
380 FORI=0TO9
390 B$= STRING$(18,32)+X$(I1,I)+
"    "+X$(I2,I)+"    "+X$(I3,I)+
"    "+X$(I4,I)
400 PRINT#-2,B$
410 NEXTI
420 PRINT#-2:PRINT#-2,"     "+ S
TRING$(70,"*"):PRINT#-2:PRINT#-2
:PRINT#-2
430 RETURN
440 PRINT#-2:PRINT#-2,A$
450 FORW=0TO4:B$=""
460 FORI=0TO2:B$=B$+"    "
470 FORJ=0TO6
480 IFYR(M+I,J,W)=0THENC$="   "E
LSEC$= STR$(YR(M+I,J,W))
490 IF LEN(C$)<3THENC$=" "+C$
500 B$=B$+C$
510 NEXTJ:NEXTI
520 PRINT#-2,B$
530 NEXTW
540 PRINT#-2:RETURN
550 CLS:PRINT TAB(6)"MICRO-80 CA
LENDAR":PRINT TAB(6)STRING$(17,"
-")
560 PRINT" THIS PROGRAM WILL GEN
ERATE A   CALENDAR FOR ANY YEAR
IN THE    RANGE 1901 - 1999. ALL
 YOU HAVE  TO DO IS TO SPECIFY T
HE YEAR!"
570 PRINT" THE PROGRAM TAKES A F
EW SECONDSTO DO THE CALCULATIONS
 AND YOU  CAN SET THE PRINTER TO
 TOP OF   FORM. WHEN EVERYTHING
IS READY, A HARDCOPY OF THE CALE
NDAR WILL BE PRINTED.":PRINT
580 A$="     SU MO TU WE TH FR S
A":A$=A$+A$+A$
590 H1$= STRING$(11,32)+"JANUARY
"+ STRING$(18,32)+"FEBRUARY"+ ST
RING$(18,32)+"MARCH"
600 H2$= STRING$(12,32)+"APRIL"+
 STRING$(21,32)+"MAY"+ STRING$(2
2,32)+"JUNE"
610 H3$= STRING$(13,32)+"JULY"+
STRING$(20,32)+"AUGUST"+ STRING$
(17,32)+"SEPTEMBER"
620 H4$= STRING$(11,32)+"OCTOBER
"+ STRING$(18,32)+"NOVEMBER"+ ST
RING$(17,32)+"DECEMBER"
630 INPUT"FOR WHICH YEAR DO YOU
WANT A    CALENDAR ";Y
640 IFY<1901ORY>1999THEN PRINT:P
RINT"OUT OF RANGE":GOTO630
```

```
650 IF 4*INT(Y/4)=Y THEN L(1)=29
660 I=Y-1901:J= INT(I/4):I=I-4*J
+2
670 K=5*(J-7* INT(J/7))+I:D=K-7*
 INT(K/7)
680 GOSUB220:PRINT:INPUT"PRESS '
ENTER' WHEN READY...";I
700 GOSUB290
710 M=0:PRINT#-2,H1$:GOSUB440
720 M=3:PRINT#-2,H2$:GOSUB440
730 M=6:PRINT#-2,H3$:GOSUB440
740 M=9:PRINT#-2,H4$:GOSUB440
750 PRINT#-2,CHR$(27)+CHR$(34):'
CLEAR ITOH EMPHASISED MODE ("H"
FOR EPSON)
760 END
```

****** HANGMAN ******
COLOUR COMPUTER

```
10 '*SUPER HANGMAN*
15 'MODIFIED FOR THE TRS80CC
20 'LOOSELY BASED ON THE PROGRAM
 IN MICRO80 JAN82 BY M.DOWNEY.
30 CLS:FOR I=1TO85:PRINT"SUPER "
;:NEXT
40 PRINT@136,STRING$(16," ");:PR
INT@232,STRING$(16," ");
50 PRINT@167,"      HANGMAN      "
;:CLEAR2500
60 PRINT@200,"  SIGHT&SOUND  ";
70 FORDL=1TO900:NEXT:CLS0
80 PRINT@420,"WELCOME TO SUPER-H
ANGMAN"
90 PRINT@448,"  YOU HAVE TO GUES
S A WORD-           ONE LETTER AT
 A TIME.";
100 SOUND150,1
110 FOR DL=1TO900:NEXT
120 GOSUB670:PRINT@456,"G O O D
 L U C K"
130 SOUND150,1
140 FOR DL=1TO500:NEXT
150 PMODE0:CLEAR600:DIMA$(13):X=
RND(TIMER)
160 GOSUB220
170 GOSUB670:PRINT@416,"   JUNIO
R OR SENIOR LEVEL?"
180 PRINT@448,"   PRESS <J> OR <
S> TO START";
190 SOUND150,1
200 LV$= INKEY$:IFLV$="J"ORLV$="
S"THEN210 ELSE 200
210 GOSUB670:GOSUB710:GOTO350
220 FOR J=1TO13
230 FORI=1TO9:READX:A$(J)=A$(J)+
CHR$(X+128):NEXT I
240 NEXT J
250 DATA25,19,19,31,31,31,19,19,
22, 48,48,53,36,32,40,58,48,48
260 DATA48,48,58,113,124,114,53,
48,48, 48,48,52,50,48,49,56,48,4
8
270 DATA32,32,47,47,32,47,47,32,
32, 32,39,46,47,47,47,45,43,32
280 DATA37,42,32,47,47,47,32,37,
42, 32,45,47,47,47,47,47,46,32
290 DATA32,32,37,47,32,47,42,32,
32, 32,32,47,47,32,47,47,32,32
300 DATA32,37,47,47,32,47,47,42,
32, 32,39,47,47,32,47,47,43,32
310 DATA32,47,47,47,47,47,47,47,
32
320 CLS0:FORI=0TO12
330 PRINT@20+I*32,A$(I+1);
340 NEXTI:RETURN
350 G$=STRING$(LEN(W$),"*"):GS$=
"":I$="":P=0
360 SOUND150,1
370 GS$=GS$+I$:PRINT@426,G$;:PRI
NT@482,GS$;:PRINT@446,"?";:PRINT
@492,"    ";
380 I$=INKEY$:IF I$=""THEN380
390 X= ASC(I$):IF(X<64)OR(X>90)T
HEN380
400 PRINT@446,I$;
410 FORI=1TO LEN(GS$)+1:IF MID$(
GS$,I,1)=I$THEN460ELSE NEXTI
420 FORI=1TOLEN(G$):IF MID$(G$,I
,1)=I$THEN480ELSE NEXT I
430 TE=0:FORI=1TO LEN(W$):IF MID
$(W$,I,1)=I$THENMID$(G$,I,1)=I$:
TE=1
440 NEXT:IF G$=W$ THEN500 ELSE I
F TE=1THEN I$="":PRINT@493,"YEP"
;:FORDL=1TO200:NEXT:SOUND200,2:G
OTO370
450 IF I=LEN(W$)+1THEN550
460 PRINT@448," YOU ALREADY TRIE
D THAT LETTER";
470 GOTO490
480 PRINT@449,"YOU'VE ALREADY GO
T THAT LETTER";
490 I$="":SOUND150,1:SOUND50,1:F
ORDL=1TO500:NEXT:GOSUB690:GOTO37
0
500 GOSUB690:PRINT@426,G$:PRINT@
456,"YOU'VE GOT IT !!";
510 PRINT@446," ";:PRINT@482,"CA
RE FOR ANOTHER GAME ??????";:FOR
I=1TO6:SOUND200,1:SOUND50,1:NEXT
520 I$=INKEY$:IFI$="Y" THEN GOSU
B670:RESTORE:RUN280
530 IFI$="N" THEN CLS:END
540 GOTO520
550 PRINT@492,"NOPE";
560 P=P+1:SOUND1,4:ON P GOTO570,
580,590,600,610,630
570 R=191:FORI=1408TO1420:POKEI,
R:NEXT:GOTO370
580 FORI=1036TO1388STEP32:POKEI,
R:NEXT:GOTO370
590 FORI=1029TO1035:POKEI,179:NE
XT:FORI=1065TO1131STEP33:POKEI,1
85:NEXT:GOTO370
600 FORI=1313TO1377STEP32:POKEI,
R:POKEI+9,R:NEXT:FORI=1314TO1321
:POKEI,188:NEXT:GOTO370
610 POKE1061,213:POKE1092,209:PO
KE1093,216:POKE1094,217:POKE1124
,218:POKE1127,218:POKE1156,212:P
OKE1157,211:POKE1158,214
620 PRINT@454,"LAST CHANCE!!!";:
FORDL=1TO200:NEXT:GOTO370
630 B=208:POKE1092,B:POKE1094,B:
POKE1124,B:POKE1156,B:POKE1127,B
:L=218:POKE1093,L:POKE1125,L:POK
E1157,L:POKE1189,219:POKE1190,L:
POKE1158,L:POKE1126,L:POKE1094,L
640 POKE1315,184:POKE1316,B:POKE
1317,B:POKE1318,B:POKE1319,181:P
OKE1351,181:POKE1383,180
650 FORI=250TO10STEP-10:SOUNDI,1
:NEXT
660 PRINT@452,"YOU'RE HUNG ON *"
W$"*";:FORDL=1TO500:NEXT:GOTO510
670 FORI=1440TO1535
680 POKEI,143:NEXT:RETURN
690 FORI=1472TO1535
700 POKEI,143:NEXT:RETURN
710 X=RND(150):IF LV$="S"THEN X=
RND(418)
720 FORI=1TOX:READW$:NEXTI
730 RETURN
740 DATAAEROPLANE,CAR,TRUCK,HORS
E,CAT,DOG,BICYCLE,HOUSE,HOLIDAY,
SCHOOL,ROAD,DOCTOR,TEACHER,NURSE
,ROOM,CLASS,PENCIL,PAPER,BABY,FA
THER,MOTHER,SISTER,BROTHER,FRIEN
D,RUBBER,ROCKET,CHICKEN,HEN
```

```
750 DATAFARM,COW,HOSPITAL,SHEEP,
BUTTER,COLOUR,YELLOW,BEDROOM,BAT
H,FOOTBALL,OCEAN,WATER,WOOL,WEAT
HER,RAIN,SNOW,SUNSHINE,HOME,SPEA
K,WITCH,WORK,WOMAN,TENNIS,TOP,TE
ACUP,BREAKFAST,LUNCH,DINNER
760 DATAFOOT,KNEE,LEG,APPLE,ORAN
GE,LEMON,QUEEN,KING,CASTLE,ARITH
METIC,BACKWARD,BALLOON,BANK,SHOP
,ENGLISH,PHOTOGRAPH,BLACKBOARD,B
LOSSOM,BOTTLE,CATERPILLAR,CAMERA
,CIRCUS,ELEPHANT,ZEBRA
770 DATADONKEY,HEADMASTER,COUNTR
Y,CRICKET,ELECTRICITY,EVENING,EX
PLORE,FAMILY,FLAVOUR,FIFTEEN,HAR
BOUR,HEART,HELICOPTER,HISTORY,PO
LICEMAN,JOURNEY,KIWI,LAUGH,LESSO
N,MARBLE,MARSHMALLOW,MISTAKE
780 DATAMOON,EARTH,MONSTER,MYSTE
RY,MUSIC,SCIENCE,NATURE,NEEDLE,N
EIGHBOUR,NUMBER,LETTER,HUNDRED,O
NION,ORCHESTRA,PIANO,PARROT,PEAC
OCK,DOLLAR,PEOPLE,PICNIC,PICTURE
,PILOT,PLANT,PUPPY,QUARTER
790 DATAQUESTION,ANSWER,RABBIT,R
ADIO,TELEVISION,TELEPHONE,RASPBE
RRY,SANDWICH,BISCUIT,SCOUT,SEVEN
,EIGHT,SHELL,SIMPLE,DAUGHTER,SUB
TRACT,SUBJECT,SUMMER,AUTUMN,SPRI
NG,HORRIBLE,TICKET,TOMORROW
800 'SENIOR
810 DATAACQUIRE,ACROBAT,ACTIVATE
,ADJACENT,ADVOCATE,AESTHETIC,AFF
LUENT,AFLAME,AGENDA,AGGRAVATE,AG
RICULTURE,AISLE,ALARM,ALBATROSS,
ALGEBRA,BALANCE,BAMBOO,BANANA,BA
SKET,BATTERY,BIBLIOGRAPHY
820 DATACALCULATE,CALENDAR,CAMOU
FLAGE,CANDIDATE,CAPACITY,CAPITAL
,CAPTIVE,CASSEROLE,CATEGORY,CENT
ENARY,CHAMPION,CHARACTER,CHEMICA
L,CHOCOLATE,CLASSICAL,CLINIC,COM
BINE,DECLARE,DECEIVE,DEGREE
830 DATADEMONSTRATE,DEPRECIATE,D
ESCEND,DEVICE,DIET,DIFFICULTY,DI
SCREET,DISPLAY,DOCUMENT,EMIGRATE
,ENGRAVE,EQUAL,ESTIMATE,EXERCISE
,FAMOUS,FALSE,FLEXIBLE,FORGIVE,F
RAGILE,GALLERY,GENEROUS
840 DATAGLANCE,GRATEFUL,GUIDE,HA
NDICAP,HARMONY,HEIR,HEPTAGON,HOL
LOW,HONOUR,COMPUTER,HYGIENE,IDLE
,ILLUSTRATE,IMPATIENCE,IMPRESS,I
NCREASE,INDICATE,INNOCENT,INTELL
ECT,INTRUDE,ITCH,JACKET
850 DATAJUNCTION,KALEIDOSCOPE,KN
OCK,LANTERN,LEAGUE,LEASH,LEOPARD
,LICENCE,LITERAL,LODGE,LOTION,MA
CHINE,MAINTAIN,MAMMOTH,MANIPULAT
E,MANUFACTURE,MARINE,MEDIUM,MAXI
MUM,MESSAGE,METEOR,MIDDLE
860 DATAMILLION,MISCHIEF,MISCELL
ANEOUS,MOMENT,MULTIPLY,MUSTARD,N
ARROW,NEPHEW,NECESSARY,NEUTRAL,N
ONSENSE,OFFICE,OPERATION,OPINION
,OVERHAUL,PADDOCK,PAINTER,PARAGR
APH,PASTE,PATROL,PEASANT
870 DATAPENSION,PERMANENT,PERSIS
T,PHYSICAL,PIECE,PILLOW,PLANET,V
ENUS,MERCURY,MARS,JUPITER,URANUS
,NEPTUNE,PLUTO,PLASTER,PLASTIC,P
LATEAU,PNEUMONIA,POISON,POPULAR,
POSITION,POVERTY,PRACTICE
880 DATAPROGRAMME,PROVIDE,PUNCTU
RE,QUANTITY,QUEUE,QUOTE,RADAR,RA
NK,RATIO,REACT,RECIPROCAL,RECOGN
ISE,REFER,REFRIGERATE,REHEARSAL,
RELATIVE,REQUIRE,RESPECT,RHYTHM,
RIDICULOUS,ROTATE,RUIN,RUSH
890 DATASABOTAGE,SADDLE,SATISFY,
SATURN,SCARLET,SCENE,SCISSORS,SC
RATCH,SCREAM,SECONDARY,SENTIMENT
,SEPERATE,SHAFT,SHINGLE,SHOULDER
,SIGNATURE,SINGLE,SLEEVE,SOCIAL,
SOUTH,SPECULATE,SPIRAL,SPLIT
900 DATASPORT,SQUEAL,STADIUM,STA
TUE,STOMACH,STRANGE,SUBSTITUTE,S
URGEON,SYMPATHY,TACTIC,TALENT,TA
RNISH,TEMPORARY,THOROUGH,THROAT,
TIMBER,TYRE,TOUCH,TRAFFIC,TREASU
RE,TRIANGLE,TROUBLE,TURTLE
910 DATAULTIMATE,UNDERSTAND,USUA
L,VAGUE,VALUE,VANISH,VARIETY,VER
SATILE,VEGETABLE,VOLUME,VISUAL,V
OWEL,WAGE,WANDER,WARRANT,WATCH,W
EIGH,WETHER,WHISPER,WHOLE,WITNES
S,WORTH,WRONG,YIELD,YOUTH
920 DATAZERO,ZENITH,WALNUT,VIEWE
R,VELOCITY,TURBULANCE,TORCH,STRE
NGTH,SPHERE,SILVER,SCORCH,SCHEDU
LE,REVOLVE,PRESCRIBE,PERIOD,ODD,
MODIFY,MATERIAL,LIQUID,JUICE,HAS
TE,FUTURE,EXTREME,DELICATE
```

****** SECTOR EDITOR ******
HITACHI PEACH

```
10 '     PROGRAM  NAME    : SECTEDT
   '     FUNCTION         : DISPLAY/EDIT D
ISK SECTORS
   '     AUTHOR           : STEVE PAYNE
20 '     CURRENT VERSION  : 01
   '     CURRENT EDIT #   : 01
   '     CREATED          : 11-Apr-82
30 '
   '               (C)1982 Integrated Digit
al Systems
   '
40 '     NOTE :  This software was writte
n for the HITACHI PEACH, and no
   '               responsibility wil be as
sumed by IDS for its use on any
   '               other machine.
   '
50 WIDTH 80 : CLS : CLEAR 4096 : DIM SEC
TOR$(16%)
 ' SECTOR$() - USED IN THE SECTOR COPY R
OUTINE - SEE LINE 1000.
60 HX$="0123456789ABCDEF": HY$=HX$+"Q"+C
HR$(28)+CHR$(29)+CHR$(30)+CHR$(31)+"U"
 : HZ$=HY$+"ZSJ"

 ' SET UP HEX TO DEC CONVERSION & COMMAN
D LOOK-UP TABLES.
70 HEADER$="SECTEDT V01-01    (C)Integrat
ed Digital Systems" : CONSOLE 0,24,0
 : PRINT HEADER$
80 PRINT : PRINT : PRINT "D - Display/Ed
it sector" : PRINT "Z - Zero sectors"
 : PRINT "C - Copy sectors" : PRINT "Q -
 Quit" : PRINT : PRINT
90 INPUT "Function ",CMD$ : ON INSTR(1%,
"DZCQ",CMD$) GOTO 110,860,980,1370
100 GOTO 50       ' INVALID OPTION.
110 LOCATE 0%,20% : INPUT "Drive, track
& sector <D,T,S> ",DRIVE%,TRACK%,SECTOR%
  : IF DRIVE%<0% OR DRIVE%>3% OR TRACK%<
0% OR TRACK%>39% OR SECTOR%<1 OR
      SECTOR%>16% THEN 110
120 CLS
130 '
140 '            READ IN THE REQUESTED SE
CTOR & DISPLAY IT IN HEX AND ASCII
150 '
160 SECT$=DSKI$(DRIVE%,TRACK%,SECTOR%)
170 OPEN "O",1,"SCRN:" : GOSUB 2000 : CL
OSE  'DISPLAY SECTOR DATA.
```

```
330 IF E% THEN E%=0% : GOTO 520          'E
%=1 - IN EDIT MODE.
340 D%=0% : D1%=0% : H0$="" : H1$="" : H
2$="" : H3$="" : LOCATE 0%,20%
  : PRINT CHR$(5); : PRINT "Command ";
350 '
360 '            OPERATIONS PERMITTED

    '  E = EDIT SECTOR     Q = QUIT     ;
= DISPLAY NEXT SECTOR
370 '  - = DISPLAY PREVIOUS SECTOR     R
= RE-DISPLAY CURRENT SECTOR
380 '  J = GET NEW DRIVE,TRACK,SECTOR NU
MBERS  P = SEND DATA TO PRINTER
    '
390 CMD$=INPUT$(1%) : IF CMD$="" THEN 39
0
  ' INPUT$() WAS USED IN PLACE OF INKEY$
 IN ORDER TO HAVE THE CURSOR DISPLAYED.
395 IF CMD$="P" THEN GOSUB 3000 : GOTO 3
90       'SEND DATA TO PRINTER.
400 IF INSTR(1%,"E;-RQJ",CMD$) THEN PRIN
T CMD$;
410 ON INSTR(1%,"E;-RQJ",CMD$) GOTO 430,
820,840,120,50,110
420 GOTO 390
430 '

    '                EDIT MODE

    '
440 PRINT CHR$(5)+"dit "; : FOR I%=1% TO
 2%
450 H0$=INPUT$(1%):IF H0$="" THEN 450 EL
SE IF INSTR(1%,HY$+"P",H0$)=0 THEN 450
460 IF H0$="Q" THEN 340          'MAY CLUTT
ER THE STACK - NOT TOO OFTEN PLEASE!!
465 IF H0$="P" THEN GOSUB 3000 : GOTO 45
0        'SEND DATA TO PRINTER.
470 H1$=H1$+H0$ : PRINT H0$; : NEXT I%
480 H2$=LEFT$(H1$,1) : H3$=RIGHT$(H1$,1)
 : H1$=""
490 D%=(INSTR(1,HX$,H2$)-1)*16+INSTR(1,H
X$,H3$)-1
500 X%=INSTR(1,HX$,H3$)-1 : Y%=INSTR(1,H
X$,H2$)-1
510 IF D%<0 OR D%>127 THEN 440 ELSE D%=D
%+1%
520 LOCATE 3*X%+6,Y%
530 H0$=INPUT$(1%): IF H0$="" THEN 530 E
LSE ON INSTR(1%,HZ$+"P",H0$)+1 GOTO 530,
    700,700,700,700,700,700,700,700,700,
700,700,700,700,700,700,700,340,540,
    580,620,660,770,1130,1260,110,535
535 GOSUB 3000 : GOTO 530          'SEND DA
TA TO PRINTER.
540 '            RIGHT ARROW KEY
550 X%=X%+1 : IF X%>15 THEN X%=0% : Y%=Y
%+1 : IF Y%>7 THEN Y%=0 : D%=0%
560 D%=D%+1 : IF D%>128 THEN D%=1 : X%=0
 : Y%=0
570 GOTO 520
580 '            LEFT ARROW KEY
590 X%=X%-1 : IF X%<0% THEN X%=15% : Y%=
Y%-1 : IF Y%<0 THEN Y%=7 : D%=129%
600 D%=D%-1 : IF D%<1 THEN D%=128 : X%=1
5 : Y%=7
610 GOTO 520
620 '            UP ARROW KEY
630 Y%=Y%-1 : IF Y%<0 THEN Y%=7 : X%=X%-
1% : D%=112%+X% : ELSE D%=D%-16%
640 IF X%<0% THEN X%=15% : D%=128%
650 GOTO 520
660 '            DOWN ARROW KEY
670 Y%=Y%+1 : IF Y%>7 THEN Y%=0 : X%=X%+
1% : D%=X%+1 : ELSE D%=D%+16%
680 IF X%>15% THEN X%=0% : D%=1%
690 GOTO 520
700 PRINT H0$; : H1$=H0$
710 H0$=INPUT$(1%) : IF H0$="" THEN 710
ELSE IF INSTR(1%,HX$,H0$)=0 THEN 710
720 '            UPDATE SECTOR AND DISPLAY
730 PRINT H0$; : H1$=H1$+H0$
740 H2$=LEFT$(H1$,1) : H3$=RIGHT$(H1$,1)
 : D1%=(INSTR(1%,HX$,H2$)-1%)*16+
        INSTR(1%,HX$,H3$)-1% : LOCATE X
%+60%,Y% : IF D1%>32 AND D1%<>255 THEN
        PRINT CHR$(D1%); ELSE PRINT "."
;
750 SECT$=LEFT$(SECT$,D%-1)+CHR$(D1%)+RI
GHT$(SECT$,128%-D%)
760 GOTO 540
770 '            UPDATE SECTOR ON DISK
780 LOCATE 0%,20% : PRINT CHR$(5)+"DISK
UPDATE - ARE YOU SURE ?? ";
790 VERIFY$=INPUT$(1) : IF VERIFY$="N" T
HEN 340 ELSE IF VERIFY$="Y" THEN 800
          ELSE 790
800 DSKO$ DRIVE%,TRACK%,SECTOR%,SECT$
810 LOCATE 0%,20% : PRINT CHR$(5)+"DISK
UPDATE COMPLETE..." : FOR I%=1 TO 1000%
  : NEXT I% : GOTO 340
820 SECTOR%=SECTOR%+1 : IF SECTOR%>16 TH
EN SECTOR%=1
  : TRACK%=TRACK%+1 : IF TRACK%>39 THEN
50
830 GOTO 850
840 SECTOR%=SECTOR%-1 : IF SECTOR%<1 THE
N SECTOR%=16
  : TRACK%=TRACK%-1 : IF TRACK%<0 THEN 5
0
850 GOTO 120
860 '
870 '            ZERO DISK SECTORS
880 '
890 PRINT "ZERO DISK SECTORS, ARE YOU CE
RTAIN?? ";
900 VERIFY$=INPUT$(1) : IF VERIFY$<>"Y"
THEN 50 ELSE PRINT VERIFY$
910 INPUT "Disk,track & sector (D,T,S) "
,DRIVE%,TRACK%,SECTOR%
920 IF DRIVE%<1 OR DRIVE%>3 OR TRACK%<0
OR TRACK%>39 OR SECTOR%<1 OR
       SECTOR%>16 THEN 910
930 INPUT "How many sectors to zero? ",C
OUNT%
940 IF COUNT%<1 OR COUNT%>(16-SECTOR%+1%
) THEN 930
950 ZERO$=STRING$(128,0)
960 FOR I%=SECTOR% TO SECTOR%+COUNT%-1%

  : DSKO$ DRIVE%,TRACK%,I%,ZERO$ : NEXT
I%
970 GOTO 50
980 '
990 '            COPY SECTORS
1000 '
1010 PRINT "COPY DISK SECTORS, ARE YOU C
ERTAIN?? ";
1020 VERIFY$=INPUT$(1) : IF VERIFY$<>"Y"
 THEN 50 ELSE PRINT VERIFY$
1030 INPUT "Source disk,track & sector <
D,T,S> ",DRIVE%,TRACK%,SECTOR%
   : IF DRIVE%<0 OR DRIVE%>3 OR TRACK%<0
 OR TRACK%>39 OR SECTOR%<1 OR
        SECTOR%>16 THEN 1030
1040 INPUT "Destination disk,track & sec
tor ",DESTD%,DESTT%,DESTS%
   : IF DESTD%<0 OR DESTD%>3 OR DESTT%<0
 OR DESTT%>39 OR DESTS%<1 OR
        DESTS%>16 THEN 1040
1050 INPUT "Sector count to transfer ";C
OUNT% : IF COUNT%<1 OR
           COUNT%>(16-SECTOR%+1%) THEN P
RINT
           "COUNT EXCEDES SOURCE MAX SEC
TOR" : GOTO 1050
1060 IF COUNT%>(16-DESTS%+1%) THEN PRINT
 "COUNT EXCEDES DESTINATION MAX SECTOR"
   : GOTO 1050
1070 FOR I%=1 TO COUNT%
```

```
1080 SECTOR$(I%)=DSKI$(DRIVE%,TRACK%,SEC
TOR%+I%-1%)
   : PRINT "Reading sector";SECTOR%+I%-1
%
   : NEXT I%
1090 PRINT : PRINT "Insert disk in drive
";DESTD%;" and hit <CR> when ready ";
   : LINEINPUT VERIFY$
1100 FOR I%=1% TO COUNT% : DSKO$ DESTD%,
DESTT%,DESTS%+I%-1%,SECTOR$(I%)
   : PRINT "Writing sector";DESTS%+I%-1%

   : NEXT I%
1110 PRINT : PRINT "Sector copy complete
" : FOR I%=1% TO 500% : NEXT I%
   : GOTO 50
1120 '
1130 '
1140 '           ZERO THE SECTOR (FROM CU
RRENT BYTE TO NOMINATED BYTE)
1150 '
1160 H0$="" : H1$="" : H2$="" : H3$="" :
 TMP$=CHR$(0)
1170 LOCATE 0%,20% : PRINT CHR$(5)+"Zero
 bytes from ";HEX$(Y%*16+X%);" to ";
1180 FOR I%=1 TO 2
1190 H0$=INPUT$(1) : IF H0$="" THEN 1190
 ELSE
        IF INSTR(1%,HX$,H0$) THEN 1200
ELSE 520
   ' IF NOT HEX CHARACTER THEN EXIT.
1200 PRINT H0$; : H1$=H1$+H0$
1210 NEXT I%
1220 H2$=LEFT$(H1$,1) : H3$=RIGHT$(H1$,1
)
   : D2%=(INSTR(1%,HX$,H2$)-1)*16+INSTR(
1%,HX$,H3$)
1230 IF D2%<D% OR D2%>128% THEN 520 ELSE
 X%=INSTR(1%,HX$,H3$)-1
   : Y%=INSTR(1%,HX$,H2$)-1
1240 SECT$=LEFT$(SECT$,D%-1)+STRING$(D2%
-D%+1%,TMP$)+RIGHT$(SECT$,128-D2%)
1250 E%=1% : CLS : GOTO 170    'RE-DISPLA
Y THIS SECTOR.
1260 '
1270 '           SUBSTITUTE NOMINATED CHA
RACTER FROM CURRENT TO SELECTED POS.
1280 '
1290 H0$="" : H1$="" : H2$="" : H3$="" :
 C0$="" : TMP$=""
1300 LOCATE 0%,20% : PRINT CHR$(5)+"Fill
 character ";
1310 FOR I%=1 TO 2
1320 C0$=INPUT$(1) : IF C0$="" THEN 1320
 ELSE IF INSTR(1%,HX$,C0$) THEN 1330
        ELSE 520
1330 PRINT C0$; : TMP$=TMP$+C0$
1340 NEXT I%
1350 C%=(INSTR(1%,HX$,LEFT$(TMP$,1))-1)*
16+INSTR(1%,HX$,RIGHT$(TMP$,1))-1
   : TMP$=CHR$(C%) : C%=0
1360 PRINT " Fill from ";HEX$(Y%*16+X%);
" to "; : GOTO 1180
1370 CLOSE : CLS : END
1999 '

     '          PRINT/DISPLAY SECTOR DAT
A
     '
2000 FOR I%=1% TO 8%
2010 LNE$=HEX$((I%-1)*16%) : IF LEN(LNE$
)<2 THEN LNE$="0"+LNE$
2020 PRINT #1%,LNE$+"    ";
2030 TMP1$=""
2040 FOR J%=1% TO 16%
2050 TMP2$=MID$(SECT$,(I%-1%)*16%+J%,1%)
2060 CHAR%=ASC(TMP2$)
2070 TMP3$=HEX$(CHAR%)+" "
2080 IF LEN(TMP3$)<3 THEN TMP3$="0"+TMP3
$
2090 PRINT #1%,TMP3$;
2100 IF CHAR%<33 OR CHAR%>125 THEN TMP2$
="."
2110 TMP1$=TMP1$+TMP2$
2120 NEXT J%
2130 PRINT #1%,"       ";TMP1$
2140 NEXT I%
2150 PRINT #1% : PRINT #1%,"DRIVE";DRIVE
%,"TRACK";TRACK%,"SECTOR";SECTOR%,
      "REL SECTOR"; : PRINT #1%,TRACK%*
16+SECTOR% : PRINT #1%
2160 RETURN
3000 '

     '          SEND SECTOR DATA TO LPT0
:
     '
3010 CLOSE : OPEN "O",1,"LPT0:"
'ASSIGN I/O CHANNEL 1 TO PRINTER.
3020 GOSUB 2000
'SEND DATA TO PRINTER.
3030 CLOSE
3040 RETURN
```

```
      **** CALENDAR ****
        HITACHI PEACH
```

```
10 '**** CALENDAR ****
20 'WRITTEN BY R.J.WIWATOWSKI
30 'NOVEMBER, 1982
40 CLEAR600
50 DEFINTD,I-N,W,Y
60 DIML(11),YR(11,6,4),X$(9,9)
70 OPEN "O",1,"LPT0:"
80 DATA31,28,31,30,31,30,31,31,30,31,30,
31
90 DATA" 000000 ","00000000","00    00",
"00    00","00    00","00    00","00
00","00    00","00000000"," 000000 "
100 DATA"    1   ","   11   ","  111   "
,"   11   ","   11   ","   11   ","   11
   ","   11   ","  111111 ","  111111 "
110 DATA" 222222 ","22222222","22    22"
,"      22","  222222","  222222 ","222
   ","22      ","22222222","22222222"
120 DATA"33333333","33333333","     33 "
,"    33  ","   333  ","    333  ","
333","33    33","33333333"," 333333 "
130 DATA"     4  ","    44  ","   444  "
,"  4444  "," 44 44  ","44  44  ","44444
444","44444444","    44  ","    44  "
140 DATA"55555555","55555555","55      "
,"55      ","5555555 ","55555555","
 55","55    55","55555555"," 555555 "
150 DATA" 666666 ","66666666","66    66"
,"66      ","6666666 ","66666666","66
 66","66    66","66666666"," 666666 "
160 DATA"77777777","77777777","      77"
,"      77","     77 ","    77  ","   77
   ","   77   ","   77   ","   77   "
170 DATA" 888888 ","88888888","88    88"
,"88    88"," 888888 "," 888888 ","88
 88","88    88","88888888"," 888888 "
180 DATA" 999999 ","99999999","99    99"
,"99    99","99999999"," 9999999","
 99","99    99","99999999"," 999999 "
190 FORI=0TO11:READL(I):NEXTI
200 FORI=0TO9:FORJ=0TO9:READX$(I,J):NEXT
J:NEXTI
210 B$=CHR$(27)+"G":PRINT#1,B$;
220 GOTO560
230 FORM=0TO11
240 W=0:DT=1
250 YR(M,D,W)=DT:DT=DT+1:D=D+1
260 IFD>6THEND=0:W=W+1:IFW>4THENW=0
270 IFDT<L(M)+1THEN250
280 NEXTM
290 RETURN
300 Q$=RIGHT$(STR$(Y),4):I1=VAL(MID$(Q$,
1,1)):I2=VAL(MID$(Q$,2,1)):I3=VAL(MID$(Q
$,3,1)):I4=VAL(MID$(Q$,4,1))
```

```
310 PRINT#1,CHR$(10);CHR$(10)
320 PRINT#1,TAB(20)"M   M III  CCC  RRRR
  000       88   00"
330 PRINT#1,TAB(20)"MM MM  I  C   C R  R
 0   0     8  8 0  0"
340 PRINT#1,TAB(20)"M M M  I  C     RRR
 0   0 ***  88  0  0"
350 PRINT#1,TAB(20)"M   M  I  C   C R  R
 0   0     8  8 0  0"
360 PRINT#1,TAB(20)"M   M III  CCC  R  R
  000       88   00"
370 PRINT#1,CHR$(10)
380 PRINT#1,"     "+STRING$(70,"*"):PRIN
T#1,CHR$(10);
390 FORI=0TO9
400 B$=STRING$(18,32)+X$(I1,I)+"    "+X$
(I2,I)+"    "+X$(I3,I)+"    "+X$(I4,I)
410 PRINT#1,B$
420 NEXTI
430 PRINT#1,CHR$(10);"     "+STRING$(70,
"*"):PRINT#1,CHR$(10);CHR$(10)
440 RETURN
450 PRINT#1,CHR$(10);A$
460 FORW=0TO4:B$=""
470 FORI=0TO2:B$=B$+"    "
480 FORJ=0TO6
490 IFYR(M+I,J,W)=0THENC$="   "ELSEC$=ST
R$(YR(M+I,J,W))
500 IFLEN(C$)<3THENC$=" "+C$
510 B$=B$+C$
520 NEXTJ:NEXTI
530 PRINT#1,B$
540 NEXTW
550 PRINT#1,CHR$(10);:RETURN
560 CLS:PRINTTAB(11)"MICRO-80 CALENDAR":
PRINTTAB(11)STRING$(17,"-"):PRINT
570 PRINT" THIS PROGRAM WILL GENERATE A
CALENDAR":PRINT"FOR ANY YEAR IN THE RANG
E 1901 - 1999.":PRINT"ALL YOU HAVE TO DO
 IS TO SPECIFY THE":PRINT"YEAR!"
580 PRINT" THE PROGRAM TAKES A FEW SECON
DS TO DO":PRINT"THE CALCULATIONS AND YOU
 CAN SET THE":PRINT"PRINTER TO TOP OF FO
RM. WHEN EVERYTHING":PRINT"IS READY, A H
ARDCOPY OF THE CALENDAR":PRINT"WILL BE P
RINTED.":PRINT
590 A$="     SU MO TU WE TH FR SA":A$=A$
+A$+A$
600 H1$=STRING$(11,32)+"JANUARY"+STRING$
(18,32)+"FEBRUARY"+STRING$(18,32)+"MARCH
"
610 H2$=STRING$(12,32)+"APRIL"+STRING$(2
1,32)+"MAY"+STRING$(22,32)+"JUNE"
620 H3$=STRING$(13,32)+"JULY"+STRING$(20
,32)+"AUGUST"+STRING$(17,32)+"SEPTEMBER"
630 H4$=STRING$(11,32)+"OCTOBER"+STRING$
(18,32)+"NOVEMBER"+STRING$(17,32)+"DECEM
BER"
640 INPUT"FOR WHICH YEAR DO YOU WANT A C
ALENDAR ";Y
650 IFY<1901ORY>1999THENPRINT:PRINT"OUT
OF RANGE":GOTO640
660 IF4*INT(Y/4)=YTHENL(1)=29
670 I=Y-1901:J=INT(I/4):I=I-4*J+2
680 K=5*(J-7*INT(J/7))+I:D=K-7*INT(K/7)
690 GOSUB230:PRINT:PRINT"PRESS 'RETURN'
WHEN READY..."
700 IFINKEY$<>CHR$(13)THEN700
710 GOSUB300
720 M=0:PRINT#1,H1$:GOSUB450
730 M=3:PRINT#1,H2$:GOSUB450
740 M=6:PRINT#1,H3$:GOSUB450
750 M=9:PRINT#1,H4$:GOSUB450
760 B$=CHR$(27)+"H"+CHR$(12):PRINT#1,B$;
770 CLOSE:END
```

****** LII/4K PROPERTY INVESTMENT ******

TRS-80/SYSTEM-80

```
10 CLS:PRINT"PROPERTY INVESTMENT SPECULATION":PRINT
20 PRINT"INPUT WHAT YOU CAN AFFORD TO REPAY
IN ORDER TO SEE WHAT YOUR MAXIMUM LOAN CAN BE"
30 PRINT:PRINTFR,:INPUT"FORTNIGHTLY REPAYMENT";FR:M=FR*2
40 PRINTWR,:INPUT"        WEEKLY RENTAL";WR:R=WR*4
50 MR=M+R:'MONTHLY REPAYMENT
60 PRINTLR,:INPUT"LENDING RATE (% PA)   ";LR:I=LR/100
70 PRINTT,:INPUT"TERM OF REPAYM. (YRS) ";T:IFT=0THEN T=10
80 P=MR*12*T/(T*I+1):'PRINCIPAL IE(MAX. LOAN)
90 PRINT"MAXIMUM LOAN YOU CAN AFFORD IS",,"$";P
100 PRINTD,:INPUT"WHAT DEPOSIT DO YOU HAVE";D
110 PD=P+D:'PRICE DEMANDED IE(MAX PRICE YOU CAN AFFORD)
120 PRINT"MAXIMUM PRICE YOU CAN AFFORD IS",,"$";PD
130 PRINT:PRINTDP,:INPUT"ENTER DESIRED PURCHASE PRICE";DP
140 MP=((DP-D)*I*T+(DP-D))/T/12:' MONTHLY PAYMENTS
150 'S.J.MANSELL.  525-5665  3 NAROOMA PL., GYMEA  2227.
160 CLS:PRINT:PRINT"PROPERTY INVESTMENT SPECULATION
========================================="
170 PRINT"FORTNIGHTLY REPAYMENT";FR,"MONTHLY REPAYM.";M
180 PRINT"WEEKLY RENTAL        ";WR,"MONTHLY REPAYM.  ";R
190 PRINT"TERM IN YEARS        ";T,"MONTHLY TOT REP.";MR
200 PRINT"LENDING RATE % PA.   ";LR
210 PRINT"MAXIMUM LOAN  $";P,"DEPOSIT";D
220 PRINT"MAXIMUM PRICE $";PD
230 PRINT
240 PRINT"DESIRED PRICE";DP,"MONTHLY PAYM.";MP
250 N=0:PRINT:INPUT"ENTER 1 FOR PRINTOUT    ELSE PRESS ENTER";N:
IFN<>1THEN 10
```

```
260 LPRINT"PROPERTY INVESTMENT SPECULATION
BY S.J. MANSELL  525-5665 1.2.81"
270 LPRINT
280 LPRINT"FORTNIGHTLY REPAYMENT";FR
290 LPRINT"WEEKLY RENTAL";WR
300 LPRINT"TOTAL MONTHLY REPAYMENT $";MR
310 LPRINT
320 LPRINT"LOAN RATE";LR;"%pa","TERM";T;"YRS"
330 LPRINT"MAX. LOAN AFFORDABLE $";P
340 LPRINT"$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$"
350 LPRINT"DEPOSIT";D
360 LPRINT"MAX. PRICE AFFORDABLE $";PD
370 LPRINT"------------------------"
380 LPRINT"DESIRED PRICE          $";DP
390 LPRINT"     MONTHLY PAYMENTS $";MP
400 LPRINT:LPRINT:LPRINT:LPRINT
410 N=0:GOTO 150
```

****** LII/16K CRICKET ******

TRS-80/SYSTEM-80

```
10 REM CRICKET BY GRAHAM  S. BULL.
20 REM P.O. BOX 95. BALHANNAH. 5242
30 REM SOUTH AUSTRALIA.
40 CLS
50 PRINT@24," =- CRICKET -= ":FORTD=1TO300:NEXTTD
60 PRINT"THERE ARE 3 GRADES OF CRICKET, THE VARIATION BEING IN B
ALL SPEED":PRINT" - THE FASTEST BEING 'ONE DAY LIMITED OVER (50
BALLS)"
70 PRINT:PRINT" - TEST CRICKET (100 BALLS)
80 PRINT:PRINT" - THE SLOWEST BEING ENGLISH CRICKET (300 BALLS)
90 PRINT:PRINT"AN INNINGS IS OVER AT THE FALL OF THE 10TH WICKET
. THE PLAYER IS           THE BATSMAN ."
100 PRINT"TO MOVE THE BAT UP 1 POSITION PRESS THE '/' KEY":PRINT
"TO LOWER THE BAT PRESS THE 'Z' KEY"
110 PRINT"THE BOWLERS GRADUALLY TIRE AS THE MATCH PROGRESSES.":P
RINT"PRESS ANY KEY TO START GAME"
120 EE$=INKEY$:IFEE$=""THEN 120  ELSE 130
130 RUN140
140 CLS
150 PRINT@465,"CRICKET";:GOSUB1080:CLS
160 PRINT@400,"WHAT GRADE OF CRICKET DO YOU PLAY ?.";:PRINT@480,
"ONE DAY MATCH    (1) ";:PRINT@544,"TEST             (2) ";:PRINT@
608,"ENGLISH          (3) ";:INPUT NN:IFNN<1ORNN>3THEN160
170 ONNNGOTO180,190,200
180 P=1:GOTO210
190 P=10:GOTO210
200 P=20:GOTO210
210 CLS:RANDOM:IFPA<20THENGOSUB1060
220 FORLP=1TO7
230 READFE,FD:IFFE=112THENRESTORE
240 FE(LP)=FE:FD(LP)=FD:SET(FE(LP),FD(LP))
250 NEXTLP
260 DATA100,20,100,41,76,33,52,34,100,5,76,7,88,12,40,16,52,41,5
2,2,64,30,76,9,100,13,112,22
270 SET(27,19):SET(27,20):SET(3,20):SET(3,22):SET(3,24)
280 GOSUB1080:CLS
290 FF=19: GG=20: HH=27: SET(HH,FF): SET(HH,GG)
300 SET(3,20): SET(3,22): SET(3,24)
310 W=RND(5): ON W GOTO 320,330,340,350,360
320 Z=4: GOTO 370
330 Z=6 : GOTO 370
340 Z=8: GOTO 370
350 Z=12 : GOTO 370
360 IF NN=1 THEN Z=24 ELSE GOTO 310  : GOTO 370
370 Y=RND(10)+16: OO=OO+1
380 IF NN=1 THEN PA=OO: IF PA>50 THEN GOTO 1050
390 IF NN=2 THEN PA=OO: IF PA>100 THEN GOTO 1050
400 IF NN=3 THEN PA=OO: IF PA>300 THEN GOTO 1020
410 V=RND(3)-2
420 T=RND(2)
430 FOR X=124 TO 52 STEP -Z
440 SET(X,Y): FOR Q= 1 TO P: NEXT Q
450 EE$=INKEY$
460 IF EE$="Z"THEN GOSUB 900
470 IF EE$="/" THEN GOSUB 950
480 RESET(X,Y)
490 Y=Y+V: IF Y<2 THEN Y=2: IF Y>45 THEN Y=45
500 IF X=52 THEN GOTO510  ELSEGOTO520
510 RESET(X,Y):GOTO 530
520 NEXT X: RESET(X,Y):
530 V=V*(-T)
540 FOR X=(52-Z)TO  4 STEP -Z
550 Y=Y+V
560 SET(X,Y): FOR Q=1 TO P: NEXT Q
570 IF POINT(4,24)=-1 THEN GOSUB 680
580 IF POINT(4,22)=-1 THEN GOSUB 680
590 IF POINT(4,20)=-1 THEN GOSUB 680
600 IF POINT(HH+1,FF)=-1 THEN GOTO 740
610 IF POINT (HH+1,GG)=-1 THEN GOTO 740
620 EE$=INKEY$
630 IF EE$="Z" THEN GOSUB 900
640 IF EE$="/" THEN GOSUB 950
650 RESET(X,Y)
660 NEXT X
670 GOSUB1080:RESET(HH,FF):RESET(HH,GG):GOTO210
680 B=B+1:PRINT@76,"BATSMAN ";B;"  OUT  ";:PRINT@200,"SCORE   "B
;" FOR  "SS;:PRINT@264,"NUMBER OF BALLS"PA;:PRINT@990,"PRESS ANY
 KEY";:GOSUB1080
690 IF B=10 THEN GOTO 1010
700 ZA$=INKEY$:IFZA$=""THEN700  ELSE710
710 P=P+5
720 RETURN
```

```
730 PRINT@140,"WELL HIT";: M=RND(6):SS=SS+M:PRINT@200,"SCORE   "
B;" FOR  "SS;:PRINT@264,"NUMBER OF BALLS"PA;:FOR TD=1 TO 700: NE
XT TD:RETURN
740 VA=V*(-1)
750 RESET(X,Y)
760 FORLP=1TO7
770 SET(FE(LP),FD(LP))
780 NEXTLP
790 FOR X=28+Z TO 124 -Z STEP Z
800 Y=Y+VA
810 IF Y<2 OR Y>45 THEN GOTO 850
820 SET(X,Y): FOR Q=1 TO P: NEXT Q
830 RESET(X,Y)
840 NEXT X
850 FORLP=1TO7
860 IFPOINT(FE(LP),FD(LP))=0THENGOTO1000
870 NEXTLP:GOSUB730
880 CLS
890 GOTO210
900 RESET(HH,FF):RESET(HH,GG)
910 FF=FF+2: GG=GG+2
920 IFGG>47THENGG=47:FF=46
930 SET(HH,FF): SET(HH,GG)
940 RETURN
950 RESET(HH,FF): RESET(HH,GG)
960 FF=FF-2: GG=GG-2
970 IFGG<1THENGG=1:FF=2
980 SET(HH,FF): SET(HH,GG)
990 RETURN
1000 PRINT@160," - CAUGHT -";:GOSUB680  :GOSUB880
1010 PRINT@140,"ALL OUT FOR ";SS;:FOR TD=1 TO 1000:NEXT TD:GOTO1
30
1020 PRINT@140,"ALL OF YOUR BATSMEN HAVE HEAT STROKE";
1030 PRINT@200,"AND HAVE RETIRED EXHAUSTED. INNINGS OVER !";
1040 PRINT@280,"YOU HAVE DECLARED FOR ";SS;:FORTD=1TO2000:NEXTTD
:GOTO130
1050 PRINT@140,"OVERS EXPIRED. YOU HAVE DECLARED FOR ";SS;:FORTD
=1TO2000:NEXTTD:GOTO130
1060 FORLP=1TO7:READFE,FD:NEXTLP
1070 RETURN
1080 FORTD=1TO500:NEXTTD:RETURN
```

****** LII/16K POTHOLE ******

TRS-80/SYSTEM-80

```
10 '            ******************
                * (C) S.CROCKETT *
                * 19 MICHELLE DR *
                * WENTWORTHVILLE *
                ******************
```

```
20 CLEAR1000:DEFINTA-Z:DIM A$(6),P$(10,2),P(24),B(4),P1(10),S(13
),02(10)
30 CLS:C$=STRING$(5,128)+CHR$(26)+STRING$(5,8)+CHR$(26)+STRING$(
5,128):A=256
40 PRINT@A,STRING$(20,155);:PRINT@A+27,STRING$(10,155);:PRINT@A+
44,STRING$(20,155);
50 A=A+64:PRINT@A,STRING$(20,166);:PRINT@A+27,STRING$(10,166);:P
RINT@A+44,STRING$(20,166);
60 A=A+64:PRINT@A,STRING$(20,153);:PRINT@A+27,STRING$(10,153);:P
RINT@A+44,STRING$(20,153);
70 IFA<>832THENA=704:GOTO40
80 DATA 128,166,191,153,128,138,171,191,151,133,160,191,131,191,
144,128,166,143,153,128,138,171,191,151,133,176,158,131,173,176
90 DATA 128,157,79,174,128,128,60,191,62,128,160,47,128,92,144,1
36,153,79,166,132,128,62,191,60,128,176,47,131,92,176
100 DATA140,179,191,179,140,128,141,188,142,128,160,191,143,191,
144,140,179,191,179,140,128,143,188,143,128,184,143,143,143,180
110 DATA128,128,128,128,128,190,179,188,179,189,187,131,151,151,
183,128,128,128,128,128,186,179,143,179,181,183,171,171,131,187
120 DATA139,180,176,184,135,160,159,175,143,180,140,135,131,163,
189,138,183,128,187,133,160,159,191,175,144,188,147,131,139,140
130 DATA128,176,128,176,128,174,188,191,188,157,140,183,131,187,
140,186,179,188,179,181,130,157,140,174,129,130,181,128,186,129
140 DATA150,179,131,179,169,149,128,79,128,170,186,143,131,143,1
81,150,179,131,179,169,149,132,45,136,170,171,183,131,187,151
150 FORO1=1TO7:FORO2=1TO2:FORJ=1TO3:FORI=1TO5
160 READX:P$(01,02)=P$(01,02)+CHR$(X):NEXTI:P$(01,02)=P$(01,02)+
CHR$(26)+STRING$(5,24):
NEXTJ,02,01:02=1:FORI=8TO10:01=RND(7):P$(I,1)=P$(01,1):P$(I,2)=P
$(01,2):NEXTI
170 B$=STRING$(5,131)
180 DATA 64,69,74,79,85,91,96,102,108,113,118,122
190 FORI=1TO12:READX:P(I)=X:NEXT
200 FORI=1TO12:P(I+12)=P(13-I)+448:NEXT
210 B(1)=277:B(2)=294:B(3)=725:B(4)=742:H=1:V=0
220 N=1:S=0:D=1:F=1
230 P(5)=-P(5):P(8)=-P(8):P(17)=-P(17):P(20)=-P(20)
240 A$(0)="*  *   P O T H O L E   **     BY  STEPHEN CROCKETT
        USE ARROW  KEYS TO MOVE THE POT-HOLE COVER  TO  TRY  A
ND  SAVE  THE  MAN  BEFORE  HE  FALLS  IN THE  HOLE.       <<PRE
SS  SKILL LEVEL TO START >> :- (0-9)    0-FAST   9-SLOW"
250 A$(1)=RIGHT$(A$(0),64)+STRING$(64,32)
260 S(1)=9:S(2)=35:S(3)=65:S(4)=120:S(5)=170:S(6)=250:S(7)=320:S
(8)=420:S(9)=560:S(10)=690:S(11)=750:S(12)=890:S(13)=1300
270 A=0:FORI=575TO512STEP-1:A=A+1:FORK=1TO20:NEXTK:K$=INKEY$:IFK
$<>""ANDK$>"/"ANDK$<":"THEN290ELSEPRINT@I,MID$(A$(0),1,A);:NEXTI
:A=0
280 FORI=1TOLEN(A$(A))-63:K$=INKEY$:IFK$<>""ANDK$>"/"ANDK$<":"TH
EN290ELSEPRINT@512,MID$(A$(A),I,64);:FORK=1TO20:NEXTK:NEXTI:IFA=
1THENA=0:GOTO270ELSEA=1:GOTO280
290 K=3:PRINT@512,RIGHT$(A$(1),64);:PRINT@512+64,RIGHT$(A$(1),64
);:FORI=1TO7:02(I)=1:NEXTI:SK=VAL(K$):E9=SK*50
300 A$(5)="HELP":A$(6)="AHHHH":A$(2)="OOPPS":A$(3)="X#$Z!":A$(4)
="OH NO"
```

```
310 FORI=1TON:FORTI=1TOE9/100+N/3*2:A$=INKEY$
320 IFA$="."THEN H=H+1:IFH=3 H=2
330 IFA$=","THENH=H-1:IFH=0 H=1
340 IFA$=CHR$(91)THEN V=V-1:IFV=-1 V=0
350 IFA$=CHR$(10)THENV=V+1:IFV=2 V=1
360 PRINT@B(B),STRING$(5,128);:PRINT@B(V*2+H),B$;:B=V*2+H
370 NEXT TI
380 PRINT@45,"S C O R E ";S;"  ";:IFS>250THENIFS1=0THENK=K+1:S1=
1
390 P1(I)=P1(I)+1:FORD=1TON:IFP1(I)=P1(D)ANDI<>DTHENP1(I)=P1(I)-
1:GOTO430ELSENEXTD
400 PRINT@P2(I),C$;:PRINT@ABS(P(P1(I))),P$(I,O2(I));:P2(I)=ABS(P
(P1(I))):IFO2(I)=1THENO2(I)=2ELSEO2(I)=1
410 IFP(P1(I))<0 GOSUB 440
420 IFP1(I)=24THENP1(I)=0
430 NEXTI:E9=E9-1:E9=E9*SGN(E9):IFSM=1THENGOSUB550:GOTO310ELSEGO
TO310
440 REM DID HE FALL ?
450 IFF>13THEN470
460 IF S<>0 AND S>S(F) ANDN<>10THEN SM=1:F=F+1:E9=SK*.6*50
470 IFP(P1(I))-192+B(B)=0THEN S=S+N:GOTO540
480 IFN<>1THENN=N-1ELSESM=0
490 P3=P2(I):R=RND(4)+1:FORO=ABS(P(P1(I)))TOABS(P(P1(I)))+320 ST
EP64
500 PRINT@P3,C$;:PRINT@0,P$(I,2);:PRINT@ABS(P(P1(I)))+320,C$;:P3
=0:PRINT@ABS(P(P1(I))),A$(R);
510 NEXTO:PRINT@ABS(P(P1(I))),"       ";
520 A1$=P$(I,1):A2$=P$(I,2):P1=P1(I):FORO1=ITO9:P$(O1,1)=P$(O1+1
,1):P$(O1,2)=P$(O1+1,2):P2(O1)=P2(O1+1):P1(O1)=P1(O1+1):NEXTO1:P
1(10)=P1:P$(10,1)=A1$:P$(10,2)=A2$:FORO1=N+1TO10:P1(O1)=0:NEXTO1
530 E9=SK*50:K=K-1:IFK=0THEN600
540 RETURN
550 IFP1(N)=13ORP1(N)=4ORP1(N)=16ORP1(N)=1ORP1(N)=10ORP1(N)=22TH
ENRETURN
560 FORO=1TON
570 IFP1(O)=1ORP1(O)=0THENRETURN
580 NEXTO
590 SM=0:N=N+1:RETURN
600 S1=0:V=0:H=1:K=3:SM=0:F=1:IFS>HSTHENHS=S
610 FORI=1TO10:PRINT@ABS(P(P1(I))),C$;:P1(I)=0:NEXTI:PRINT@600,"
G A M E   O V E R";:PRINT@10,"HIGH SCORE";HS;:S=0:N=1:GOTO270
```

****** LII/16K m/1 FASTER ******

TRS-80/SYSTEM-80

```
00100
00110
00120
00130  ;_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_
00140
00150  ;Program to alter baud rate of all BASIC cassette
00160  ;operations. (SYSTEM not included as BASIC operation.)
00170  ;Most saving on baud rate from altering data format.
00180  ;Baud rate is altered by using a '#' directly before
00190  ;the normal BASIC cassette operation e.g. #CLOAD.
00200  ;All operations will function as normal without the
00210  ;preceeding '#' sign. Two extra functions are included:-
00220  ;#ON will turn the motor of a specified cassette on
00230  ;until any key is pressed.
00240  ;#LIST will list the BASIC lines which contain BASIC
00250  ;cassette operations.
00260
00270  ;SPECIAL NOTE:- No memory need be reserved for this
00280  ;program as it reserves it's own. The only requirements
00290  ;are that the ORiGin is set to an address above that
00300  ;held in (40A4H) and that already reserved memory
00310  ;is at a point greater than the final loading address
00320  ;of this program + 623T bytes.
00330
00340
00350  ; Written by J. Langsford.
00360  ; PO BOX 118 WOODSIDE
00370  ; OCTOBER 1982
00380
00390  ;_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_
00400
00410
00420  ;ORiGin set at 7000H but could be anywhere as this
00430  ;program reserves it's own memory. A higher ORiGin
00440  ;is suggested for disk based systems so as not to
00450  ;interfere with BASIC programs but it does not matter
00460  ;provided the DOS does not extend beyond 7000H
00470
00480         ORG     7000H   ;SET TO 8000H FOR DISK SYSTEMS
00490
00500  ;Original entry point to put in right spot
00510
00520 ENTRY   LD      BC,ZLO-START+1 ;BC=length of program
00530         LD      HL,(40B1H) ;HL=already reserved memory
00540         OR      A       ;Reset carry
00550         SBC     HL,BC   ;Get new reserve address
00560         LD      DE,ZLO+1 ;Get end of this tape
00570         RST     18H     ;See if HL too low
00580         JP      C,197AH ;?OM Error if too small
00590         LD      (40B1H),HL ;Set new reserved
00600         PUSH    HL      ;Save it
00610         LD      DE,OFFCEH ;DE=-50
00620         ADD     HL,DE   ;Calc string boundary
00630         LD      (40A0H),HL ;Store string boundary
00640         POP     HL      ;Restore new reserved to HL
00650         PUSH    HL      ;Save again
00660         LD      DE,START ;DE=old start address
00670         OR      A       ;Reset carry
00680         SBC     HL,DE   ;Calculate displacement
00690         PUSH    BC      ;Store number of bytes to shift
```

```
00700           PUSH    HL      ;HL to stack
00710           POP     BC      ;BC=displacement
00720           LD      IY,TABLE ;Get table of addresses to change
00730 LPC       LD      L,(IY)  ;Get LSB of address
00740           LD      A,L     ;Put in A also
00750           INC     IY      ;Shift to MSB
00760           OR      (IY)    ;Combine to see if finished
00770           JR      Z,FIN   ;Go if finished
00780           LD      H,(IY)  ;Get MSB of address
00790           INC     IY      ;Point to next for later
00800           LD      E,(HL)  ;LSB of that address to E
00810           INC     HL      ;Point to MSB
00820           LD      D,(HL)  ;Get MSB
00830           EX      DE,HL   ;Exchange for 16-bit arith
00840           ADD     HL,BC   ;Calc new value for address
00850           EX      DE,HL   ;Put DE to HL for LD r,(HL) inst.
00860           LD      (HL),D  ;Put MSB of new address
00870           DEC     HL      ;Point to LSB
00880           LD      (HL),E  ;Store LSB
00890           JR      LPC     ;Go back for next address
00900 FIN       LD      HL,START ;Get old start address
00910           POP     BC      ;Get number of bytes to shift
00920           POP     DE      ;Get Destination
00930           LDIR            ;Shift the program
00940           LD      HL,ENTRY ;Test to see if this tape has
00950           LD      DE,(40F9H) ;interfered with possible BASIC
00960           RST     18H     ;program in memory
00970           PUSH    AF      ;Save flags
00980           LD      HL,(40B1H) ;Get new entry address
00990           PUSH    HL      ;Save it
01000           LD      HL,MESS ;Not a mess but a message
01010           CALL    2B75H   ;Print it.
01020           POP     HL      ;Restore it
01030           PUSH    HL      ;Save it again
01040           CALL    0FAFH   ;Print the new ENTRY POINT
01050           CALL    20F9H   ;Video to new line
01060           POP     HL      ;Restore it again
01070           POP     AF      ;Restore flags
01080           PUSH    HL      ;Save ENTRY as RET address
01090           JP      C,1B4DH ;Has interfered so reset all BASIC
01100                           ;pointers and variables.
01110           JP      1B61H   ;Reset only variable pointers
01130 MESS      DEFW    1F1CH   ;Clear screen
01140           DEFM    '1000 BAUD ROUTINE ENABLE' ;Message text
01150           DEFB    0DH                        ;New line
01160           DEFM    'Written by J.Langsford'
01170           DEFB    0DH
01180           DEFM    'ENTRY POINT = '
01190           DEFB    0       ;Message terminator
01210
01220
01230  ;Table of values of address which alter depending on
01240  ;where in memory this program is supposed to go.
01250
01260 TABLE     DEFW    START+1 ;The addresses in this table
01270           DEFW    LPB+1   ;are CALLS, and JUMPS in the
01280           DEFW    PRINT+2 ;original program which must
01290           DEFW    PRINT+5 ;be altered to suit whatever
01300           DEFW    PRNT+9  ;location in memory this
01310           DEFW    PRNT+0CH ;program is to be loaded.
01320           DEFW    PRNTC+1 ; The entries in this
01330           DEFW    PRNTC+7 ;table are used as pointers
01340           DEFW    STORE+1 ;to addresses in the program
01350           DEFW    SWAP+4  ;which are loaded and then
01360           DEFW    INPUT+0AH ;increased by the amount
01370           DEFW    LP0+1   ;of the shift to the new
01380           DEFW    G0+28H  ;memory location of the
01390           DEFW    G0+2DH  ;program. They are then
01400           DEFW    LP9+3   ;returned to their place in
01410           DEFW    LP9+8   ;the original program before
01420           DEFW    LP9+0BH ;the shift to the new location.
01430           DEFW    LP9+0FH
01440           DEFW    LP5+3
01450           DEFW    LP5+0AH
01460           DEFW    LP5+0DH
01470           DEFW    LP5+10H
01480           DEFW    CNT2+0FH
01490           DEFW    LOOK+3
01500           DEFW    LOOK+0CH
01510           DEFW    NEXT+3
01520           DEFW    CNT3+6
01530           DEFW    LP7+1
01540           DEFW    WRITE2+1
01550           DEFW    LP4+1
01560           DEFW    LP1+1
01570           DEFW    BIT+1
01580           DEFW    LP6+1
01590           DEFW    NXTT+6
01600           DEFW    NXTT+10H
01610           DEFW    LDELAY+1
01620           DEFW    0
01630
01640  ;Macro for short delays
01650
01660 SDLY      MACRO   #CT
01670           LD      B,#CT   ;Get count value into B
01680           DJNZ    $       ;Hang around for a while
01690           ENDM
01700
01710  ;Original entry to set up RST 10 new call
01720
01730 START     LD      HL,G0 ;HL=start of RST 10 processing
01740           LD      (4004H),HL ;Put in RST 10 RAM area
01750           JP      6CCH    ;Back to BASIC
01760
01770  ;Print on cassette at faster rate
01780
01790 PRINT     RST     10H     ;Get next character
01800           CALL    WRITEL  ;Write leader etc.
01810           CALL    SWAP    ;Swap old DOS exit for new
```

```
01820          LD      A,80H    ;Set out put device
01830          LD      (409CH),A ;Store output device #
01840          JP      209BH    ;Back to ROM
01850
01860  ;Entry from DOS exit before printing a character
01870
01880 PRNT     INC     SP       ;Remove DOS return address
01890          INC     SP
01900          LD      A,(409CH) ;Get otuput device
01910          OR      A        ;Set flags
01920          LD      A,C      ;Restore character to be printed
01930          POP     BC       ;Restore BC
01940          JP      M,PRNTC  ;Go print if cassette
01950          CALL    SWAP     ;Restore DOS exit if not
01960          JP      32AH     ;Then go back to print character
01970 PRNTC    CALL    WRITE    ;Print to cassette
01980          CP      0DH      ;See if output finished
01990          RET     NZ       ;Return if not finished
02000          CALL    SWAP     ;Restore DOS exit if finished
02010          RET
02020
02030  ;Storage area for DOS exit
02040
02050 STORE    JP      PRNT
02060
02070  ;Subroutine to exchange DOS exits
02080
02090 SWAP     EXX              ;Save registers
02100          DI               ;Interupts not wanted
02110          PUSH    AF       ;Save A
02120          LD      HL,STORE ;HL= storage area for DOS exits
02130          LD      DE,41C1H ;DE= Dos exit to swap
02140          LD      B,3      ;B= 3 bytes to swap
02150 LP8      LD      A,(DE)   ;Get byte from exit
02160          LD      C,(HL)   ;Get byte from storage
02170          LD      (HL),A   ;Store old byte from exit
02180          LD      A,C      ;A=old byte from storage
02190          LD      (DE),A   ;Store it
02200          INC     HL       ;Bump to next byte in store
02210          INC     DE       ;Bump to next byte in exit
02220          DJNZ    LP8      ;Loop till 3 bytes swapped
02230          POP     AF       ;Restore A
02240          EXX              ;Restore registers
02250          EI               ;Interupts OK now
02260          RET
02270
02280  ;Routine to turn on cassette motor until key pressed
02290
02300 ONC      RST     10H      ;Get next character
02310          CALL    1FEH     ;Turn on cassette desired
02320          CALL    49H      ;Wait for key
02330          JP      1F8H     ;Go turn off cassette and return
02340
02350  ;Input from cassette
02360
02370 INPUT    CALL    2828H    ;Error if direct statement
02380          RST     10H      ;Get next character
02390          LD      A,0      ;Set input from cassette
02400          LD      (40A9H),A ;Store INPUT source flag
02410          CALL    LOOKL    ;Find sync byte
02420          PUSH    HL       ;Save HL
02430          LD      B,0FAH   ;B = max length of INPUT
02440          LD      HL,(40A7H) ;HL= address of I/O buffer
02450 LP0      CALL    READ     ;Get byte from cassette
02460          LD      (HL),A   ;Store it in buffer
02470          INC     HL       ;Bump to next place in buffer
02480          CP      0DH      ;Test for end of input
02490          JP      Z,21BDH  ;Go if input finished
02500          DJNZ    LP0      ;Read rest of input
02510          JP      21BDH    ;Back to ROM when finished
02520
02530  ;Entry point from RST 10 call
02540
02550 GO       EX      (SP),HL  ;Get ret address to HL & save HL
02560          PUSH    DE       ;Save DE
02570          LD      DE,1D5BH ;DE=possible return address
02580          RST     18H      ;Compare DE with HL
02590          POP     DE       ;Restore DE
02600          EX      (SP),HL  ;Restore HL & return address
02610          JP      NZ,1D78H ;Go if not return looking for
02620          POP     DE       ;Clear RST return address
02630          RST     10H      ;Get next token
02640          LD      DE,1D1EH ;DE = return address
02650          PUSH    DE       ;Push ret address to stack
02660          RET     Z        ;Return if end of statement
02670          CP      '#'      ;Test for # sign
02680          JP      NZ,1D60H ;Go if no #
02690          INC     HL       ;Point to token following # sign
02700          LD      A,(HL)   ;Get next character
02710          CP      89H      ;Test for INPUT
02720          JR      Z,INPUT  ;Go if INPUT
02730          CP      0A1H
02740          JR      Z,ONC
02750          CP      0B9H     ;Test for CLOAD
02760          JR      Z,CLOAD  ;Go if CLOAD
02770          CP      0B4H     ;Test for LIST
02780          JP      Z,LIST   ;Go if LIST
02790          CP      0B2H     ;Test for PRINT
02800          JP      Z,PRINT  ;Go if PRINT
02810          CP      0BAH     ;Test for CSAVE
02820          JP      NZ,1997H ;SN Error if # & no valid token
02830
02840  ;New CSAVE routine
02850
02860          RST     10H      ;Get next token
02870          LD      B,1
02880          JR      NC,CSAVE ;Csave if not a digit following
02890          CALL    1E5AH    ;Get number of copies
02900          LD      A,D      ;MSB to A
02910          OR      A        ;Set flags
```

```
02920           CALL    2B22H   ;Test for > 255
02930           OR      A
02940           JR      Z,CSAVE
02950           LD      B,A     ;Number of copies to B
02960 CSAVE     PUSH    BC      ;Save number of copies
02970           CALL    1FEH    ;Turn on tape
02980           CALL    2337H   ;Evaluate file name
02990           POP     BC      ;Restore number of copies
03000           PUSH    HL      ;Save execution point
03010           PUSH    BC      ;Save number of copies again
03020           CALL    2A13H   ;Address of file name to DE
03030           POP     BC      ;Restore copies
03040 LP9       PUSH    BC
03050           PUSH    DE      ;Save file name address
03060           CALL    LWRITE  ;Write a leader
03070           LD      A,0D3H  ;Pointer to BASIC program
03080           CALL    WRITE2  ;Write three of them
03090           CALL    WRITE
03100           LD      A,(DE)  ;File name to A
03110           CALL    WRITE   ;Write file name
03120           LD      HL,(40A4H) ;HL = program pointer
03130           EX      DE,HL   ;Program pointer to DE
03140           LD      HL,(40F9H) ;Get end of Program pointer
03150 LP5       LD      A,(DE)  ;Get a byte from program
03160           INC     DE      ;Bump pointer to next byte
03170           CALL    WRITE   ;Write byte from program
03180           RST     18H     ;Test for end of program HL=DE
03190           JR      NZ,LP5 ;Continue writing if not at end
03200           POP     DE      ;Restore file address
03210           CALL    LDELAY  ;Delay between copies
03220           CALL    LDELAY
03230           CALL    LDELAY
03240           POP     BC      ;Get number of copies
03250           DJNZ    LP9     ;Loop till all copies written
03260           JP      2C1AH   ;Back to ROM
03270
03280  ;New CLOAD routine
03290
03300 CLOAD     RST     10H     ;Get next token
03310           CALL    1FEH    ;Turn on cassette
03320           LD      A,(HL)  ;Get next byte to execute
03330           SUB     0B2H    ;Test for ? i.e. #CLOAD?
03340           JR      Z,CLQ   ;Jump to #CLOAD? routine
03350           XOR     A       ;Zero A
03360           DEFB    1       ;Dummy statement
03370 CLQ       DEFB    2FH     ;CPL A=-1 (0FFH) if #CLOAD?
03380           DEFB    23H     ;INC HL Position to file name
03390           PUSH    AF      ;Save #CLOAD? flag
03400           DEC     HL      ;Back space HL
03410           RST     10H     ;Get next byte
03420           LD      A,0     ;Set A for no name
03430           JR      Z,CNT2  ;Jump if no name
03440           CALL    2337H   ;Evaluate name
03450           CALL    2A13H   ;Name pointer to DE
03460           LD      A,(DE)  ;Get file name
03470 CNT2      LD      L,A     ;Save file name
03480           POP     AF      ;Restore #CLOAD? flag
03490           OR      A       ;Set flags
03500           LD      H,A     ;Save #CLOAD? flag
03510           LD      (4121H),HL ;Save flags and file name
03520           CALL    Z,1B4DH ;Call NEW routine if not #CLOAD?
03530           LD      HL,(4121H) ;Restore File name and flags
03540           EX      DE,HL   ;Save in DE
03550           CALL    LPB     ;Find sync byte
03560 LOOK      LD      B,3     ;Look for 3 0D3 bytes
03570           CALL    READ    ;Read a byte
03580           SUB     0D3H    ;Test for BASIC pointer
03590           JR      NZ,LOOK ;Go back if not three 0D3 bytes
03600           DJNZ    LOOK+2  ;Read three of them
03610           CALL    READ    ;Read file name
03620           INC     E       ;Test to see if user specified
03630           DEC     E       ;       file name.
03640           JR      Z,CNT4  ;Jump if no name specified
03650           CP      E       ;See if correct file name
03660           JR      NZ,CNT3 ;Jump wrong file name read
03670 CNT4      LD      HL,(40A4H) ;Get start of program table
03680 NEXT      LD      B,3     ;Set B to number of zeros to read
03690           CALL    READ    ;Read a byte
03700           LD      E,A     ;Save byte read
03710           SUB     (HL)    ;Test to see if same as in memory
03720           AND     D       ;Test for #CLOAD?
03730           JP      NZ,2C8AH ;Print BAD if #CLOAD? & not same
03740           LD      (HL),E  ;Save the byte read
03750           CALL    196CH   ;Test for out of memory
03760           LD      A,(HL)  ;Get byte read
03770           OR      A       ;Test for end of statement
03780           INC     HL      ;Bump to next storage location
03790           JR      NZ,NEXT ;Go back to read next byte
03800           CALL    22CH    ;Blink an asterisk
03810           DJNZ    NEXT+2  ;Go back till three zeros read
03820           JP      2C77H   ;Back to ROM for finale
03830
03840  ;Skip over this file if file name not the same as read.
03850
03860 CNT3      LD      (3C3EH),A ;Save file name
03870           LD      B,3     ;B=3bytes to search for
03880           CALL    READ    ;Read a byte
03890           OR      A       ;Test for zero
03900           JR      NZ.CNT3+3 ;Go look for a zero
03910           DJNZ    CNT3+5  ;Got a zero but find 3 of them
03920 LP7       CALL    READ    ;Look for next program
03930           CP      0A5H    ;Test for sync
03940           JR      NZ,LP7 ;No so keep looking
03950           JR      NEXT    ;Got one so go read it.
03960
03970  ;Write two bytes the same
03980
03990 WRITE2    CALL    WRITE
04000           JR      WRITE
04010
```

```
04020  ;Write leader and sync byte routine
04030
04040 WRITEL   CALL    1FEH     ;Turn on correct cassette
04050 LWRITE   XOR     A        ;Zero A
04060          LD      B,80H    ;Don't really need 256 zeros.
04070 LP4      CALL    WRITE    ;Write a zero
04080          DJNZ    LP4      ;Do 256 of them
04090          LD      A,0A5H   ;Ld A with sync byte and fall
04100                           ; through to write it.
04110
04120  ;1000 baud write routine
04130
04140 WRITE    PUSH    HL       ;Save registers
04150          PUSH    BC
04160          PUSH    DE
04170          PUSH    AF
04180          DI              ;Do not disturb
04190          LD      D,A      ;Save byte to write
04200          LD      C,9      ;Count of pulses to write
04210          SCF              ;Set carry to write sync pulse
04220 LP1      CALL    BIT      ;Write a pulse if carry set
04230          LD      A,D      ;Restore rotated byte to write
04240          DEC     C        ;Decrement count of bits
04250          RRA              ;Rotate bit to write into carry
04260          LD      D,A      ;Save rotated byte
04270          JR      NZ,LP1 ;Go back if count not zero
04280          SDLY    12H      ;Delay between bytes
04290          EI               ;Listening now
04300          JP      279H     ;Let ROM handle POP's and return
04310 BIT      JP      C,PULSE ;Write a pulse if carry
04320          SDLY    74H      ;Delay if no bit to write
04330          LD      HL,(0)   ;Wait extra 16 T-states
04340          RET
04350
04360  ;Look for sync byte routine
04370
04380 LOOKL    CALL    1FEH     ;Turn on cassette
04390 LPB      CALL    READ     ;Read a byte
04400          OR      A        ;Set flags and make sure a zero
04410          JR      NZ,LPB ;read first so tape can speed up
04420 LP6      CALL    READ     ;Read a byte
04430          CP      0A5H     ;Test for sync byte
04440          JR      NZ,LP6 ;Loop if no sync byte
04450          LD      A,'#'    ;A=Ascii '#'
04460          LD      (3C3EH),A ;Put in corner of screen
04470          LD      A,'*'    ;A=Ascii '*'
04480          LD      (3C3FH),A ;Put in corner of screen
04490          RET
04500
04510  ;1000 baud read routine
04520
04530 READ     PUSH    HL       ;Save registers
04540          PUSH    BC
04550          PUSH    DE
04560          DI              ;Go away I'm busy
04570          LD      C,8      ;Load count with 8 bits to read
04580 LP2      IN      A,(0FFH) ;Look for clock pulse
04590          RLA              ;Test if clock pulse
04600          JR      NC,LP2 ;Go look again if no pulse
04610          SDLY    41H      ;Delay before flip-flop is reset
04620 LP3      CALL    21EH     ;Reset the flip-flop
04630          SDLY    76H      ;Delay before reading pulse
04640          IN      A,(0FFH) ;Read possible pulse
04650          RLA              ;Rotate bit read into carry
04660          RR      D        ;Rotate carry into D
04670          DEC     C        ;Count of bits read
04680          INC     HL       ;Delay for 6 T states
04690          SET     1,H      ;Delay 8 more T states
04700          JR      NZ,LP3 ;Go back if bit count NZ
04710          CALL    21EH     ;Again reset flip-flop
04720          LD      A,D      ;Get byte read to A
04730          EI               ;Busy no longer
04740          JP      27AH     ;Let ROM handle POP's and return
04750
04760  ;Print lines with cassette function.
04770
04780 LIST     XOR     A        ;A=0 for screen
04790          LD      (409CH),A ;Set output device to screen
04800          CALL    20F9H    ;Make sure at begginning of line
04810          RST     10H      ;Get next token
04820          PUSH    HL       ;Save execution address
04830          LD      HL,(40A4H) ;HL=start of BASIC program
04840 NXTL     LD      A,(HL)   ;Get LSB of line pointer
04850          INC     HL       ;Bump to MSB
04860          OR      (HL)     ;Test for end of program
04870          JR      NZ,CNT1 ;Go if not end
04880          POP     HL       ;Restore execution address
04890          RET
04900 CNT1     INC     HL       ;Bump to LSB of line number
04910          LD      E,(HL)   ;Get LSB of line number to E
04920          INC     HL       ;Bump to MSB
04930          LD      D,(HL)   ;DE=line number
04940          INC     HL       ;Go into statement
04950 NXTT     LD      A,(HL)   ;Get a token
04960          INC     HL       ;Get ready for next token
04970          OR      A        ;See if end of line
04980          JR      Z,NXTL   ;Go back to do next line if end
04990          CALL    COMP     ;Look for cassette token
05000          JR      Z,TESTN ;Go look for # if token< B9
05010          CP      '#'      ;Look for # sign
05020          JR      NZ,NXTT ;Not token or # so go back
05030          LD      A,(HL)   ;Get character in A
05040          CALL    COMP     ;Test for cassette function
05050          JR      Z,SKP    ;Go print line no if right token
05060          JR      NXTT     ;Go back for next character if not
05070 TESTN    CP      0B9H     ;Set flags for <> B9
05080          JR      NC,SKP   ;Go print line if CSAVE or CLOAD
05090          DEC     HL       ;Backspace so RST checks this one
05100          RST     10H      ;Get next character
```

```
05110           LD      A,(HL)  ;Get character to A
05120           CP      '#'     ;Test for # after PRINT or INPUT
05130           JR      NZ,NXTT ;Go back for next if not #
05140 SKP       EX      DE,HL   ;Line number to HL
05150           PUSH    DE      ;Save current address on stack
05160           CALL    0FAFH   ;Print line oumber
05170           LD      A,20H   ;Blank to A
05180           CALL    32AH    ;Print a blank between numbers
05190           POP     HL      ;Restore current address
05200
05210  ;Skip over the rest of this line
05220
05230 LPA       LD      A,(HL)  ;Get token to A
05240           INC     HL      ;Skip to next token
05250           OR      A       ;Set flags
05260           JR      NZ,LPA ;Go back ig not end of this line
05270           JR      NXTL    ;Go back to do next line
05280
05290  ;Write a pulse on tape
05300
05310 PULSE     LD      HL,0FC01H ;Mask to set positive pulse
05320           CALL    221H    ;Go set pulse
05330           SDLY    0BH     ;Load B with delay
05340           LD      HL,0FC02H ;Mask to set negative swing
05350           CALL    221H    ;Go do it
05360           SDLY    0BH     ;Load B with delay
05370           LD      HL,0FC00H ;Set end of pulse
05380           CALL    221H    ;Set output to centre
05390           SDLY    4CH     ;Delay again
05400           RET
05410
05420  ;Test for PRINT, INPUT, CLOAD or CSAVE tokens
05430
05440 COMP      CP      0B9H    ;Test for CLOAD token
05450           RET     Z       ;Return with zero if CLOAD
05460           CP      0BAH    ;Test for CSAVE
05470           RET     Z       ;Return with zero if CSAVE
05480           CP      89H     ;Test for INPUT
05490           RET     Z       ;Return with zero if INPUT
05500           CP      0B2H    ;Test for PRINT (set flags)
05510           RET
05520
05530  ;Longer delay routine
05540
05550 LDELAY    CALL    DELAY   ;Extra long delay
05560 DELAY     PUSH    BC      ;Save BC
05570           LD      BC,0    ;Delay count to BC
05580           CALL    60H     ;Call delay in ROM
05590           POP     BC      ;Restore BC
05600 ZLO       RET
05610
05620  ;All over bar the shouting. That wasn't too difficult
05630  ;now was it ?
05640
05650           END     ENTRY   ;Here's where it's at
```

```
7000:  01 6F 02 2A B1 40 B7 ED 42 11 6A 73 DF DA 7A 19
7010:  22 B1 40 E5 11 CE FF 19 22 A0 40 E1 E5 11 FB 70
7020:  B7 ED 52 C5 E5 C1 FD 21 B1 70 FD 6E 00 7D FD 23
7030:  FD B6 00 28 10 FD 66 00 FD 23 5E 23 56 EB 09 EB
7040:  72 2B 73 18 E5 21 FB 70 C1 D1 ED B0 21 00 70 ED
7050:  5B F9 40 DF F5 2A B1 40 E5 21 70 70 CD 75 2B E1
7060:  E5 CD AF 0F CD F9 20 E1 F1 E5 DA 4D 1B C3 61 1B
7070:  1C 1F 31 30 30 30 20 42 41 55 44 20 52 4F 55 54
7080:  49 4E 45 20 45 4E 41 42 4C 45 0D 57 72 69 74 74
7090:  65 6E 20 62 79 20 4A 2E 4C 61 6E 67 73 66 6F 72
70A0:  64 0D 45 4E 54 52 59 20 50 4F 49 4E 54 20 3D 20
70B0:  00 FC 70 A8 72 06 71 09 71 1C 71 1F 71 25 71 2B
70C0:  71 2F 71 35 71 5D 71 66 71 9C 71 A1 71 CA 71 CF
70D0:  71 D2 71 D6 71 E2 71 E9 71 EC 71 EF 71 21 72 26
70E0:  72 2F 72 3E 72 5D 72 65 72 6E 72 79 72 89 72 9A
70F0:  72 AE 72 06 73 10 73 5F 73 00 00 21 74 71 22 04
7100:  40 C3 CC 06 D7 CD 72 72 CD 31 71 3E 80 32 9C 40
7110:  C3 9B 20 33 33 3A 9C 40 B7 79 C1 FA 24 71 CD 31
7120:  71 C3 2A 03 CD 7F 72 FE 0D C0 CD 31 71 C9 C3 13
7130:  71 D9 F3 F5 21 2E 71 11 C1 41 06 03 1A 4E 77 79
7140:  12 23 13 10 F7 F1 D9 FB C9 D7 CD FE 01 CD 49 00
7150:  C3 F8 01 CD 28 28 D7 3E 00 32 A9 40 CD A4 72 E5
7160:  06 FA 2A A7 40 CD BF 72 77 23 FE 0D CA BD 21 10
7170:  F4 C3 BD 21 E3 D5 11 5B 1D DF D1 E3 C2 78 1D D1
7180:  D7 11 1E 1D D5 C8 FE 23 C2 60 1D 23 7E FE 89 28
7190:  C2 FE A1 28 B4 FE B9 28 5E FE B4 CA E8 72 FE B2
71A0:  CA 04 71 FE BA C2 97 19 D7 06 01 30 0C CD 5A 1E
71B0:  7A B7 CD 22 2B B7 28 01 47 C5 CD FE 01 CD 37 23
71C0:  C1 E5 C5 CD 13 2A C1 C5 D5 CD 75 72 3E D3 CD 6D
71D0:  72 CD 7F 72 1A CD 7F 72 2A A4 40 EB 2A F9 40 1A
71E0:  13 CD 7F 72 DF 20 F8 D1 CD 5E 73 CD 5E 73 CD 5E
71F0:  73 C1 10 D3 C3 1A 2C D7 CD FE 01 7E D6 B2 28 02
7200:  AF 01 2F 23 F5 2B D7 3E 00 28 07 CD 37 23 CD 13
7210:  2A 1A 6F F1 B7 67 22 21 41 CC 4D 1B 2A 21 41 EB
7220:  CD A7 72 06 03 CD BF 72 D6 D3 20 F7 10 F7 CD BF
7230:  72 1C 1D 28 03 BB 20 1F 2A A4 40 06 03 CD BF 72
7240:  5F 96 A2 C2 8A 2C 73 CD 6C 19 7E B7 23 20 EC CD
7250:  2C 02 10 E9 C3 77 2C 32 3E 3C 06 03 CD BF 72 B7
7260:  20 F8 10 F8 CD BF 72 FE A5 20 F9 18 CE CD 7F 72
7270:  18 0D CD FE 01 AF 06 80 CD 7F 72 10 FB 3E A5 E5
7280:  C5 D5 F5 F3 57 0E 09 37 CD 99 72 7A 0D 1F 57 20
7290:  F7 06 12 10 FE FB C3 79 02 DA 33 73 06 74 10 FE
72A0:  2A 00 00 C9 CD FE 01 CD BF 72 B7 20 FA CD BF 72
72B0:  FE A5 20 F9 3E 23 32 3E 3C 3E 2A 32 3F 3C C9 E5
72C0:  C5 D5 F3 0E 08 DB FF 17 30 FB 06 41 10 FE CD 1E
72D0:  02 06 76 10 FE DB FF 17 CB 1A 0D 23 CB CC 20 EE
72E0:  CD 1E 02 7A FB C3 7A 02 AF 32 9C 40 CD F9 20 D7
72F0:  E5 2A A4 40 7E 23 B6 20 02 E1 C9 23 5E 23 56 23
7300:  7E 23 B7 28 EF CD 52 73 28 0C FE 23 20 F2 7E CD
7310:  52 73 28 0D 18 EA FE B9 30 07 2B D7 7E FE 23 20
7320:  DF EB D5 CD AF 0F 3E 20 CD 2A 03 E1 7E 23 B7 20
7330:  FB 18 C1 21 01 FC CD 21 02 06 0B 10 FE 21 02 FC
7340:  CD 21 02 06 0B 10 FE 21 00 FC CD 21 02 06 4C 10
7350:  FE C9 FE B9 C8 FE BA C8 FE 89 C8 FE B2 C9 CD 61
7360:  73 C5 01 00 00 CD 60 00 C1 C9
```

***** NEXT MONTH'S ISSUE *****

Next month's issue will contain at least the following programs plus the usual features and articles. An (80) after a program title indicates that the program will be for TRS-80 Model 1/3 or System 80/Video Genie computers. (Colour) indicates that the program will be for the TRS-80 Colour Computer and the Hitachi Peach.

** MATRIX MANIPULATIONS LI/4K (80) **

This is the same program as that published in the May 1982 issue for Level 2 machines. It has been modified by the author to run in a Level 1. The Level 1 version provides all of the functions that the Level 2 version had.

** FLEXITIME LII/4K (80) **

This is for all you public servant types, or anyone else who works on the system of flexible hours. Did you ever lose track of how many hours you had stored up? Well, this program will help to get rid of the problem.

** MOVE BY 1's LII/16K (80) **

On the face of it, this program appears to be a very simple game requiring you to move through a maze to get out, in the least possible number of moves. The only problem is that it is not so simple, as you will see next month.

** BIORHYTHM (COLOUR) **

Your biorhythms are the cycles that are related to how you feel and can have a bearing on what happens in your life. With this program, you can check your life cycles as often as you please.

** CODE BREAKER LII/4K (80) **

This program was written specifically to solve those cryptograms (secret messages in a simple substitution code), which appear in magazines. Remember such things as Captain Marvel's code rings? Well, this program will help you crack the code.

** CHECKSUM LII/4K-48K (80) **

Now you can add a checksum to a BASIC program. This program tells you what the length and checksum should be for any BASIC program loaded into memory. You then enter those figures into the program itself and save it. Next time you load the program, you can compare the checksums to see if you got a good load.

** LOAN CALCULATION PACKAGE (COLOUR) **

Are you concerned with how fluctuating interest rates can affect your housing loan? This program provides Repayment calculation, Remaining term calculation, Remaining balance calculation, Dissection of repayments and Repayment factor calculations.

APPLICATION FOR PUBLICATION OF A PROGRAM IN MICRO-80

Date .....

To **MICRO-80** SOFTWARE DEPT. PO BOX 145 MORPHETT VALE SA 5162

Please consider the enclosed program for . . .

Tick where appropriate

(i) Publication in MICRO-80 .....

(ii) Publication on disk or cassette only .....

(iii) Both .....

Name .....

Address .....

.....

.....Postcode .....

***** CHECK LIST *****

Please ensure that the cassette or disk is clearly marked with your name and address, program name(s), Memory size, Level I, II, System 1 or 2, Edtasm, System, etc. The use of REM statements with your name and address is suggested, in case the program becomes separated from the accompanying literature.

Ensure that you supply adequate instructions, notes on what the program does and how it does it, etc.

For system tapes, the start, end, and entry points, etc.

The changes or improvements that you think may improve it.

Please package securely — padabags are suggested — and enclose stamps or postage if you want your cassette or disk returned.

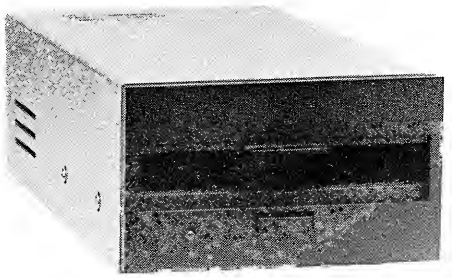
***** CASSETTE/DISK EDITION INDEX *****

The cassette edition of MICRO-80 contains all the software listed each month, on cassette. The cassette also contains the source code for machine language programs which may not have been printed due to space restrictions. All programs are recorded twice. Level I programs can only be loaded into a Level I TRS-80 if the Level I in Level 2 program from the MICRO-80 Software Library - Vol. 1 is first loaded into your Level 2 TRS-80 or System 80/Video Genie. Note: System 80/Video Genie computers have had different tape-counters fitted at different times. The approximate start positions shown are correct for the very early System 80 without the volume control or level meter. They are probably incorrect for later machines. The rates for a cassette subscription are printed on the inside front cover of each issue of the magazine.

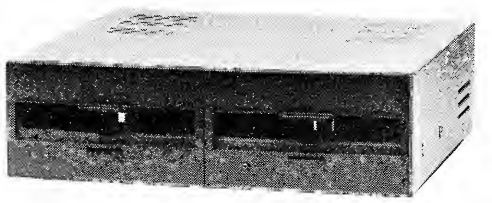
The disk edition contains all those programs which can be executed from disk, including Level I programs. Level I disk programs are saved in the NEWDOS format. Users require the Level I/CMD utility supplied with NEWDOS+ or NEWDOS 80 version 1.0 to run them.

SIDE 1	TYPE	I.D.	DISK FILESPEC	APPROX. START POSITION CTR-41	CTR-80	SYSTEM 80
POTHOLE	LII/16K	P	POTHOLE/BAS	18	10	5
"	"	"	"	64	36	17
CRICKET	LII/16K	C	CRICKET/BAS	107	60	32
"	"	"	"	142	80	43
PROPERTY INVESTMENT	LII/16K	I	PROPERTY/BAS	178	100	57
"	"	"	"	197	111	64
FASTER	SYSTEM	FASTER	FASTER/CMD	217	122	68
"	"	"	"	229	129	74
"	EDTASM	"	FASTER/EDT	242	136	78
SIDE 2						
FASTER	EDTASM	FASTER	FASTER/EDT	18	10	5

SAVE A PACKET ON MICRO-80's DISK DRIVE PACKAGES FOR TRS-80 MODEL 1 AND SYSTEM 80 MICROCOMPUTERS



SINGLE DRIVE PACKAGE *from* ... $499



DUAL DRIVE PACKAGE *from* ... $874

Bigger volume means lower cost price, which we are passing on to you. Avoid the annoying bundle of cables, wires and separate boxes. MICRO-80 is now offering our well-proven MPI disk drives in attractive, self-contained single or dual-drive cabinets complete with internal power supply. Our drive Ø and dual-drive packages also include the appropriate version of DOSPLUS and dual-drive cable.

The best news of all is the specially reduced package prices ...
SAVE $23 — $107 over our already low prices!

Choose the appropriate system from the table below:

DRIVE TYPE	No. of Tracks	No. of Heads	Capacity	Dosplus Version	Price	* Saving
DRIVE Ø						
1 x MPI B51	40	1	100K	3.3	$499	$77.95
1 x MPI B52	40	2	200K	3.4	$639	$97.95
1 x MPI B92	80	2	400K	3.4	$799	$107.95
DRIVE 1						
1 x MPI B51	40	1	100K	—	$415	$23.00
1 x MPI B52	40	2	200K	—	$525	$23.00
1 x MPI B92	80	2	400K	—	$695	$23.00

*Represents the saving compared with buying all the items included in the package separately

•Drive Ø package includes one bare disk drive, self-contained single-drive cabinet/power supply as illustrated, two drive cable and the version of DOSPLUS indicated.

•Drive 1 package includes one bare disk drive and self-contained single-drive cabinet/power supply as illustrated.

If it's a dual-drive system you need, then take advantage of our dual-drive package and
SAVE a further $40 on the price of two single-drive packages ...

DRIVE TYPE	No. of Tracks	No. of Heads	Capacity	Dosplus Version	Price
2 x MPI B51	40 ea	1 ea	2 x 100K	3.3	$874
2 x MPI B52	40 ea	2 ea	2 x 200K	3.4	$1125
2 x MPI B92	80 ea	2 ea	2 x 400K	3.4	$1454

Dual-drive package includes two bare disk drives, self-contained dual-drive cabinet/power supply as illustrated, two drive cables and the version of Dosplus indicated.

NOTE: All 40 track drives are completely compatible with 35 track operating systems such as TRSDOS. DOSPLUS allows you to realise an additional 14% capacity compared with TRSDOS. Under DOSPLUS 3.4, 80 track drives can read 35/40 track diskettes.

All disk drive components are still available separately:

BARE DRIVES — MPI drives offer the fastest track-to-track access time (5 milliseconds) available. All drives are capable of operating in double density for 80% greater storage capacity.

	Price	Freight
MPI B51 40 track, single-head, 100K	$349	$5.00
MPI B52 40 track, dual-head, 200K	$449	$5.00
MPI B92 80 track, dual-head, 400K	$619	$5.00
Separate, dual-drive power supply	$85	$8.00

	Price	Freight
Self-contained, single drive cabinet/power supply	$99	$5.00
Self-contained, dual-drive cabinet/power supply	$135	$5.00
Two drive cable	$39	$2.00
Fan drive cable	$49	$2.00
DOSPLUS 3.3	$99.95	$2.00
DOSPLUS 3.4	$149.95	$2.00

Prices are FOB Adelaide. Add $5.00 freight for single drive package, $10.00 for dual-drive package. Prices are in Australian dollars. Freight is road freight anywhere in Australia.

All items carry a 90-day parts and labour warranty. Repairs to be carried out in our Adelaide workshops.

LEVEL 2 ROM

ASSEMBLY LANGUAGE TOOLKIT

by Edwin Paay

FOR TRS-80 MODEL 1, MODEL 3 AND SYSTEM 80/VIDEO GENIE

This is a new package consisting of two invaluable components:

- **A ROM REFERENCE** Manual which catalogues, describes and cross-references the useful and usable ROM routines which you can incorporate into your own machine language or BASIC programs.
- **DBUG**, a machine language disassembling debugging program to speed up the development of your own machine language programs. DBUG is distributed on a cassette and may used from disk or cassette.

Part 1 of the ROM REFERENCE manual gives detailed explanations of the processes used for arithmetical calculations, logical operations, data movements etc. It also describes the various formats used for BASIC, System and Editor/Assembly tapes. There is a special section devoted to those additional routines in the TRS-80 Model 3 ROM. This is the first time this information has been made available, anywhere. Differences between the System 80/Video Genie are also described. Part 1 is organised into subject specific tables so that you can quickly locate all the routines to carry out a given function and then choose the one which meets your requirements.

Part 2 gives detailed information about each of the routines in the order in which they appear in the ROM. It describes their functions, explains how to use them in your own machine language programs and notes the effect of each on the various Z80 registers.

Part 2 also details the contents of system RAM and shows you how to intercept BASIC routines. With this knowledge, you can add your own commands to BASIC, for instance, or position BASIC programs in high memory — the only restriction is your own imagination!

The Appendices contain sample programmes which show you how you can use the ROM routines to speed up your machine language programs and reduce the amount of code you need to write.

DBUG: Eddy Paay was not satisfied with any of the commercially available debugging programs, so he developed his own. DBUG: allows you to single-step through your program; has a disassembler which disassembles the next instruction before executing it or allows you to bypass execution and pass on through the program, disassembling as you go; displays/edits memory in Hex or ASCII; allows Register editing; has the ability to read and write System tapes and all this on the bottom 3 lines of your screen, thus freeing the rest of the screen for program displays. Four versions of DBUG are included in the package to cope with different memory sizes.

The best news of all is the price. The complete Level 2 ROM ASSEMBLY LANGUAGE TOOLKIT is only:

— Aus. $29.95 + $2.00 p&p
— UK £18.00 + £1.00 p&p

SPECIAL OFFER TO OWNERS OF THE LEVEL II ROM REFERENCE MANUAL ...
UPGRADE TO THIS ASSEMBLY LANGUAGE TOOKIT FOR ONLY $19.95!
Send back your original Level II ROM Reference Manual plus a cheque, money order or Bankcard authorisation for $19.95 plus $2.00 p&p and we will send you the new ASSEMBLY LANGUAGE TOOLKIT